École Supérieure d'informatique et du Commerce

Déploiement Automatisé d'un Système d'Exploitation avec Application sur un Réseau de Machines Clients

Le projet a été conduit par **Said Hashemi** sous la direction de **M. El Alama**, en date du 03/05/2024.

Objectifs

Dans le cadre de mon projet, je vise à comprendre pleinement le processus de déploiement d'un système d'exploitation avec une application sur une machine cliente, tout en acquérant des compétences pratiques dans l'utilisation d'outils spécifiques à cette tâche. Je souhaite automatiser ce déploiement pour réduire le temps et minimiser les erreurs, tout en personnalisant les installations pour répondre aux besoins spécifiques du projet. Mon objectif est également de documenter méticuleusement chaque étape, des configurations aux tests et validations, afin de fournir une référence précieuse pour les futurs déploiements et de faciliter le partage des connaissances au sein de mon équipe.

Je m'engage à réaliser des tests rigoureux pour garantir que le déploiement fonctionne correctement dans divers contextes et sur différentes configurations de machines clientes, assurant ainsi un déploiement réussi et fiable à chaque fois. En atteignant ces objectifs, je développerai des compétences pratiques précieuses dans le domaine du déploiement logiciel, renforçant ainsi mon expertise et ma compréhension des processus essentiels en informatique.

Table des matières

Objectifs .....
Les Solutions Possibles 1
Choix de la solution WDS et MDT pour le déploiement système 2
Configuration du serveur 3
Guide pour Télécharger et Installer les Outils Requis 4
Configuration de Deployment Share 6
Déploiement 8
Création d'une tâche de séquence 9
Solution pour quelques erreurs existantes dans Microsoft MDT 11
Quelques Modifications Utiles 13
Manipulation sur le service WDS 15
Installation de Windows 10 avec FileZilla sur le PC Client 16
Déploiement de FileZilla via MDT 22
Résultat 25

Les Solutions Possibles

Il existe plusieurs solutions possibles pour déployer un logiciel et un système d'exploitation vers des machines clientes à partir d'un serveur. Par exemple :

1. **PXE Boot (Preboot Execution Environment) :** Avec PXE Boot, les machines clientes peuvent démarrer à partir du réseau et se connecter à un serveur PXE pour télécharger et exécuter un système d'exploitation. Nous pouvons configurer le serveur PXE pour déployer une image système personnalisée contenant le système d'exploitation et le logiciel.
2. **Outils de gestion de configuration :** Des outils comme Ansible, Puppet, Chef et SaltStack permettent d'automatiser le déploiement et la gestion des logiciels et des configurations sur les machines clientes. Nous pouvons écrire des playbooks ou des recettes pour spécifier comment installer et configurer le logiciel et le système d'exploitation sur les machines clientes.
3. **Solutions de déploiement d'images :** Des outils tels que Clonezilla, Fog Project, WDS (Windows Deployment Services) ou WPKG (Windows Package Manager) nous permettent de créer et de déployer des images système sur les machines clientes. Nous préparons une image système avec le système d'exploitation et le logiciel préinstallé, puis nous la déployons sur les machines clientes.
4. **Containers et orchestration :** Utilisons des technologies de conteneurisation comme Docker ou Kubernetes pour créer des conteneurs contenant le logiciel requis. Nous pouvons ensuite orchestrer le déploiement de ces conteneurs sur les machines clientes à l'aide de solutions d'orchestration comme Kubernetes, Docker Swarm ou Nomad.
5. **Virtualisation et provisioning :** Utilisons des solutions de virtualisation comme VMware, VirtualBox, Hyper-V ou KVM pour créer des machines virtuelles avec le système d'exploitation et le logiciel préinstallé. Nous pouvons ensuite provisionner ces machines virtuelles sur les machines clientes à l'aide de scripts ou d'outils de gestion de virtualisation.
6. **Scripting personnalisé :** Nous pouvons également créer vos propres scripts personnalisés pour automatiser le déploiement du logiciel et du système d'exploitation sur les machines clientes. Nous utilisons des langages de script comme Bash, PowerShell, Python ou Ruby pour écrire des scripts qui effectuent les tâches de déploiement nécessaires.

Choix de la solution WDS et MDT pour le déploiement système

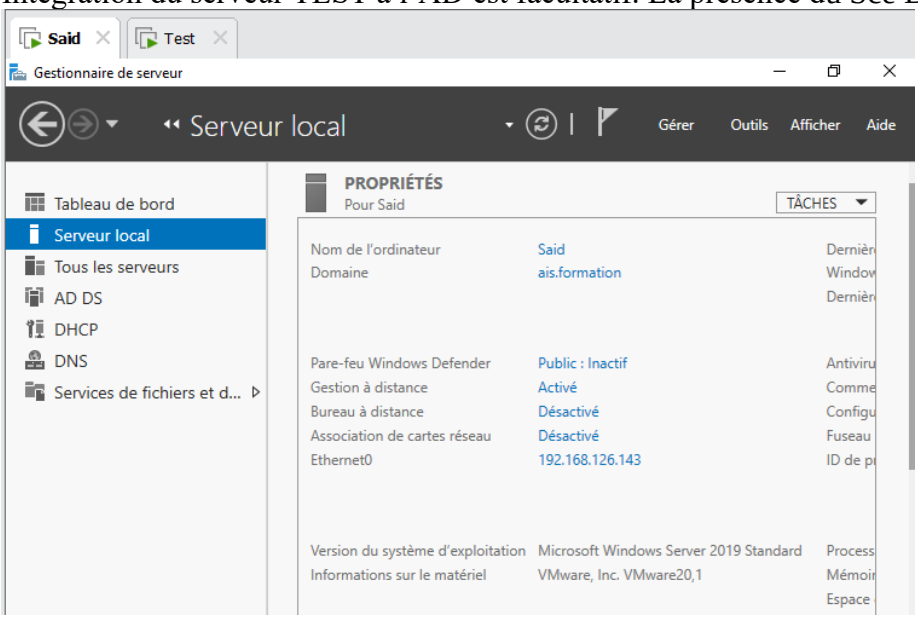
J'ai opté pour l'utilisation de Windows Deployment Services (WDS) et Microsoft Deployment Toolkit (MDT) en raison de leur intégration transparente avec l'écosystème Windows existant. Cette solution offre une compatibilité maximale avec notre infrastructure, facilitant ainsi son intégration et son maintien. De plus, MDT permet une personnalisation avancée des images système, nous offrant la flexibilité nécessaire pour inclure les pilotes matériels, les mises à jour logicielles et les applications spécifiques à nos besoins, garantissant ainsi une expérience utilisateur homogène sur toutes les machines clientes.

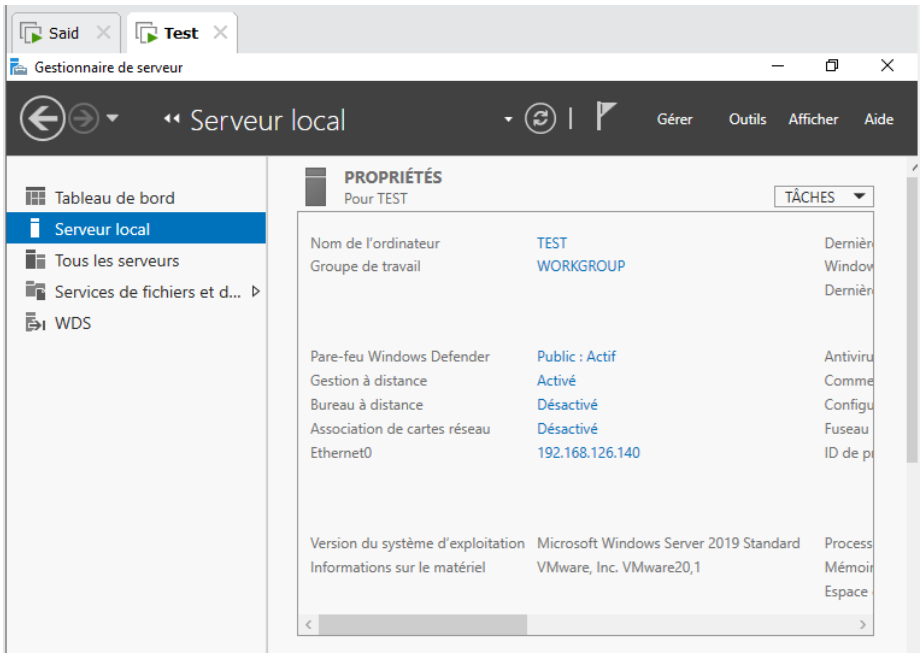
L'automatisation du déploiement est un autre facteur déterminant dans le choix de cette solution. Grâce à MDT, nous pouvons créer des séquences de tâches personnalisées pour automatiser l'installation du système d'exploitation, des applications et des paramètres système. Cela nous permet de réduire significativement le temps et les efforts nécessaires pour déployer de nouvelles machines, tout en garantissant la cohérence et la conformité de nos déploiements. En outre, la gestion centralisée des images et des configurations à travers WDS et MDT simplifie la maintenance et la mise à jour de notre infrastructure, assurant ainsi sa stabilité et sa fiabilité à long terme.

Configuration du serveur

J'ai un serveur nommé Said qui joue actuellement les rôles d'Active Directory (AD) et de serveur DNS. Pour utiliser les fonctionnalités WDS, j'ai décidé d'installer Windows Deployment Services (WDS) sur un autre serveur nommé TEST. WDS va permettre de faciliter le déploiement des images système sur les machines clientes de manière automatisée. En intégrant cette fonctionnalité à mon serveur Said, je centralise davantage les services essentiels de notre infrastructure, ce qui simplifie la gestion et la maintenance globales.

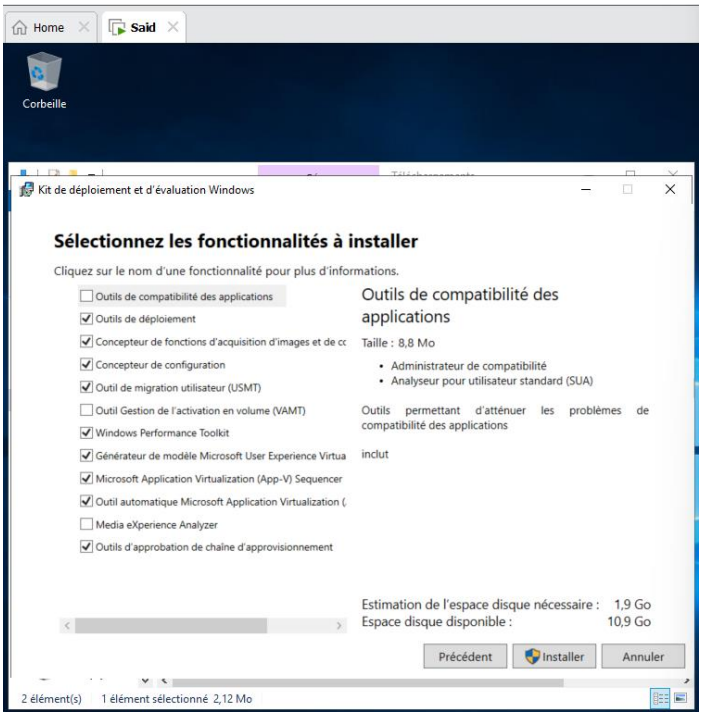
Pour installer le rôle WDS sur mon serveur nommé Test, je commence par ouvrir le Gestionnaire de serveur. Ensuite, je clique sur "Ajouter des rôles et des fonctionnalités" et je sélectionne "Services de déploiement Windows". Après avoir suivi l'assistant d'installation et configuré les paramètres initiaux via le Gestionnaire de services de déploiement, le serveur est prêt à déployer des images système sur les machines clientes de manière automatisée.
Intégration du serveur TEST à l'AD est facultatif. La présence du Sce DCHP est obligatoire.



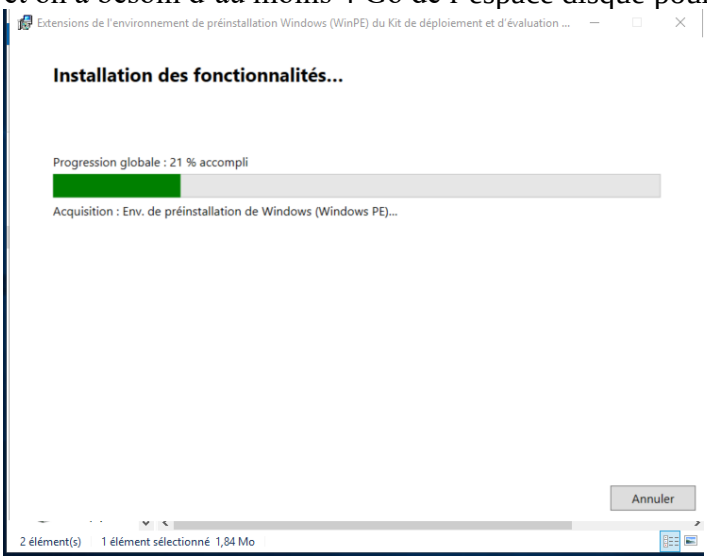


Guide pour Télécharger et Installer les Outils Requis

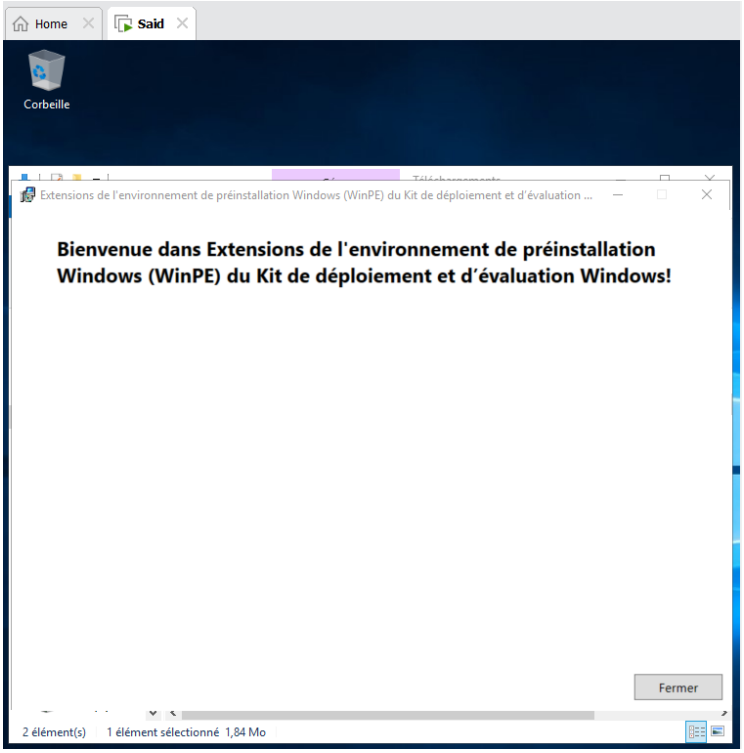
Je vais ouvrir le lien indiqué ci-dessous pour télécharger Windows ADK et Windows PE Add-on :
https://learn.microsoft.com/en-us/windows-hardware/get-started/adk-install
Une fois le Kit de déploiement et d'évaluation Windows téléchargé, nous procédons à sa configuration. Après avoir accepté le contrat de licence, une fenêtre comme celle ci-dessous s'affiche, nous permettant de sélectionner les fonctionnalités à installer, ce que j'ai fait. Ensuite, nous cliquons sur le bouton "Installer".



Il peut y avoir des soucis de scripts avec cette version de Windows ADK (sortie en septembre 2023) pour résoudre les soucis, consultez le lien ci-dessous :
https://learn.microsoft.com/en-us/windows-hardware/get-started/what-s-new-in-kits-and-tools?source=recommendations#windows-pe-known-issues?WT.mc_id=AZ-MVP-5004580
Maintenant on doit installer WINPE, il faut noter qu'on l'installe à l'emplacement par défaut, et on a besoin d'au moins 4 Go de l'espace disque pour effectuer cette installation.



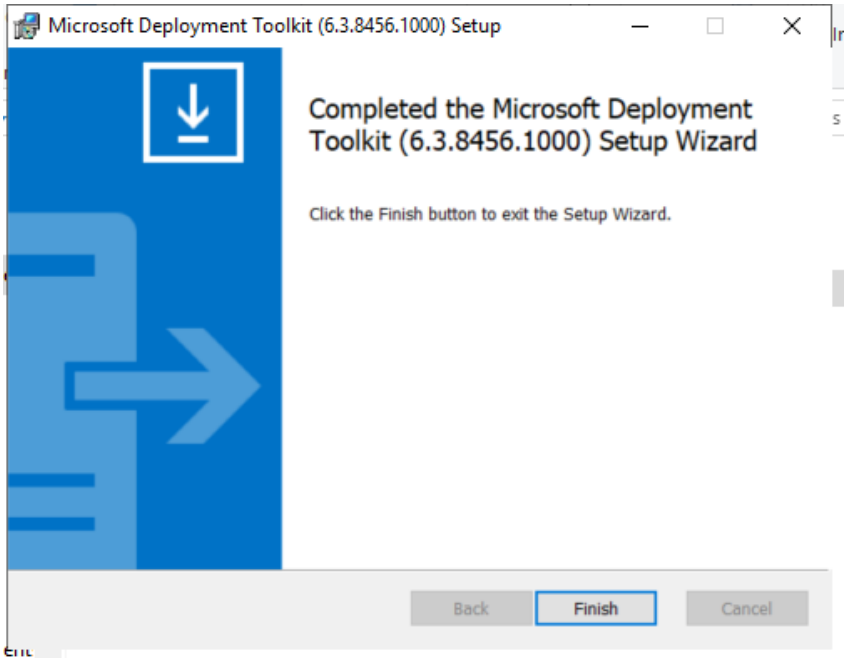
Nous avons réussi l'installation du WINPE avec succès.



Maintenant, il est temps d'installer le dernier composant nécessaire : le composant MDT. Vous trouverez ci-dessous le lien de téléchargement pour l'installation :

https://www.microsoft.com/en-sa/download/details.aspx?id=54259

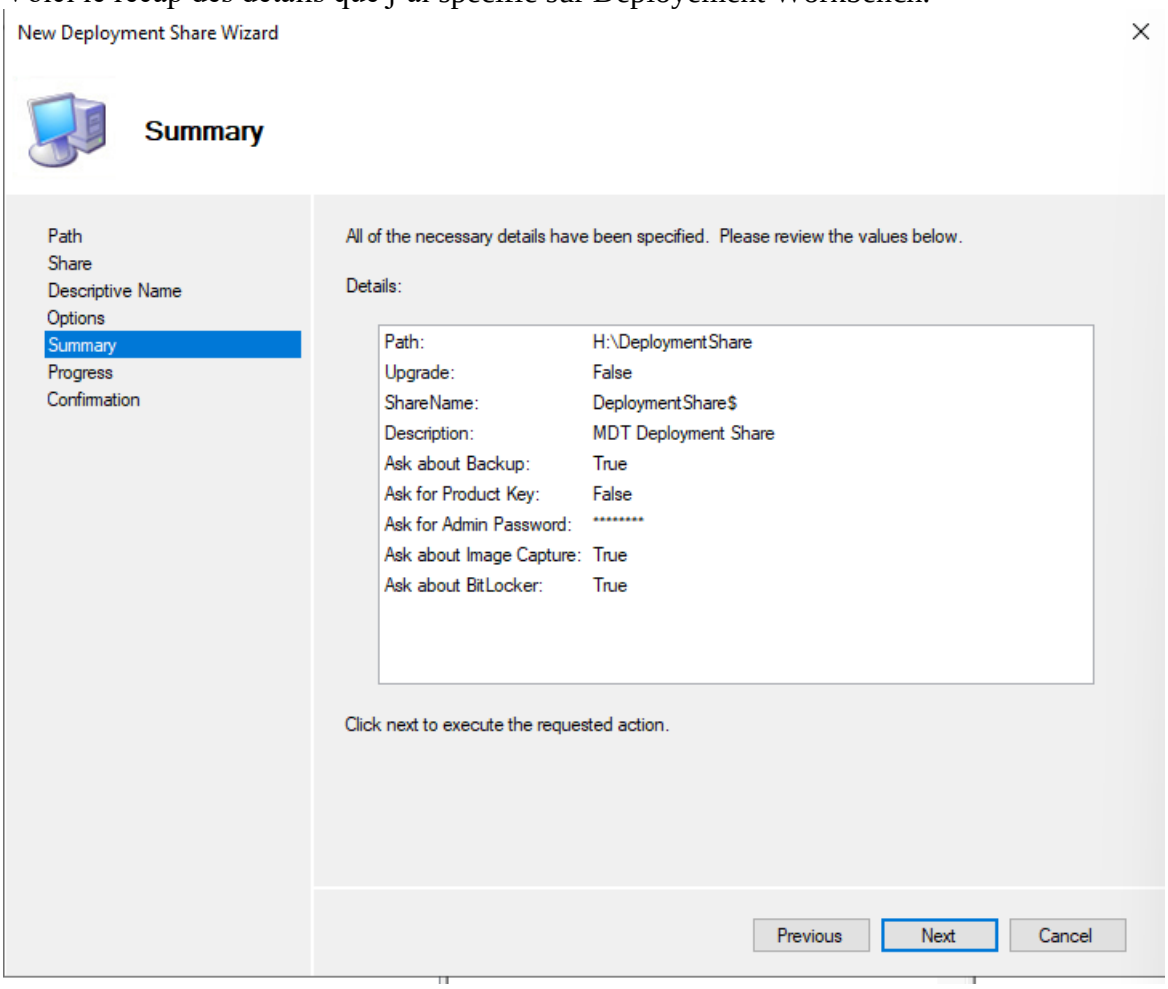
Et là j'ai réussi à installer le Windows Déploiement Toolkit :



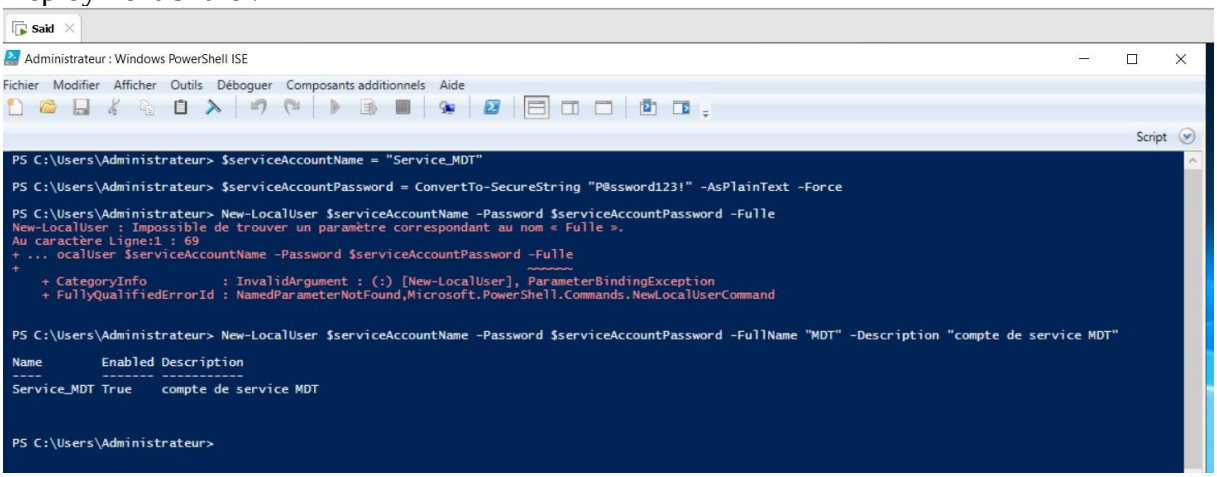
Configuration de Deployment Share

Nous ouvrons la console dédiée nommée "Deployment Workbench". Ensuite il faut créer un « Deployment Share ». J'ai créé un lecteur nommé "WDS" pour stocker les données associées à ce déploiement.

Voici le récap des détails que j'ai spécifié sur Deployement Workbench.



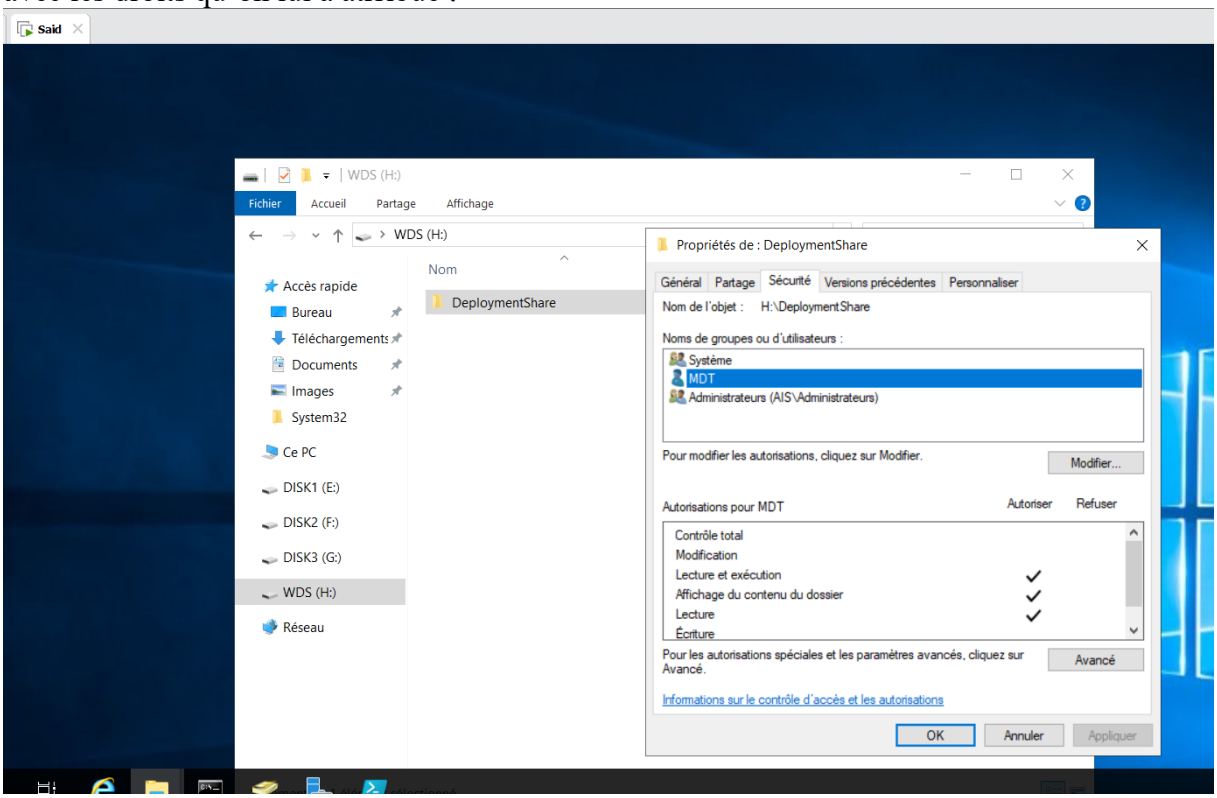
Pour accéder au Delpoyment Share, je crée un utilisateur « Service_MDT » par ligne de commandes PowerShell, pour que cet utilisateur aussi aie le droit de lecture sur le Deployment Share :



Maintenant on doit attribuer des droits appropriés à l'utilisateur « Service_MDT » :

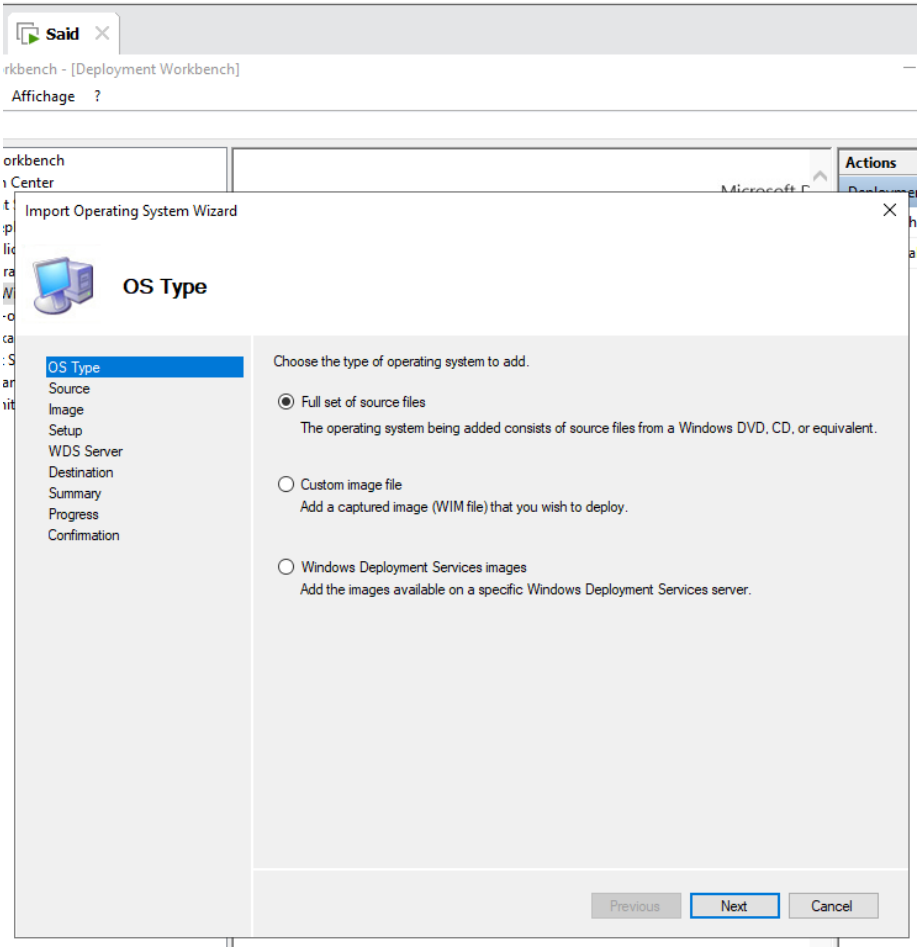
```
PS C:\Users\Administrateur> Grant-SmbShareAccess -Name "DeploymentShare$" -AccountName "Service_MDT" -AccessRight Read -Force

Name             ScopeName AccountName            AccessControlType AccessRight
----             --------- -----------            ----------------- -----------
DeploymentShare$ *         BUILTIN\Administrateurs Allow             Full
DeploymentShare$ *         AIS\Service_MDT         Allow             Read


PS C:\Users\Administrateur> $MDTSharePath = "\\SAID\DeploymentShare$"

PS C:\Users\Administrateur> $Acl= Get-Acl $MDTSharePath

PS C:\Users\Administrateur> $Rule =New-Object System.Security.AccessControl.FileSystemAccessRule("Service_MDT", "ReadAndExecute", "ContainerInherit, objectInherit", "None", "Allow")

PS C:\Users\Administrateur> $Acl.SetAccessRule($Rule)

PS C:\Users\Administrateur> set-Acl $MDTSharePath $Acl

PS C:\Users\Administrateur> |
```

Dans les propriétés du Deployment Share on voit bien que l'utilisateur MDT a bien été ajouté avec les droits qu'on lui a atrribué :

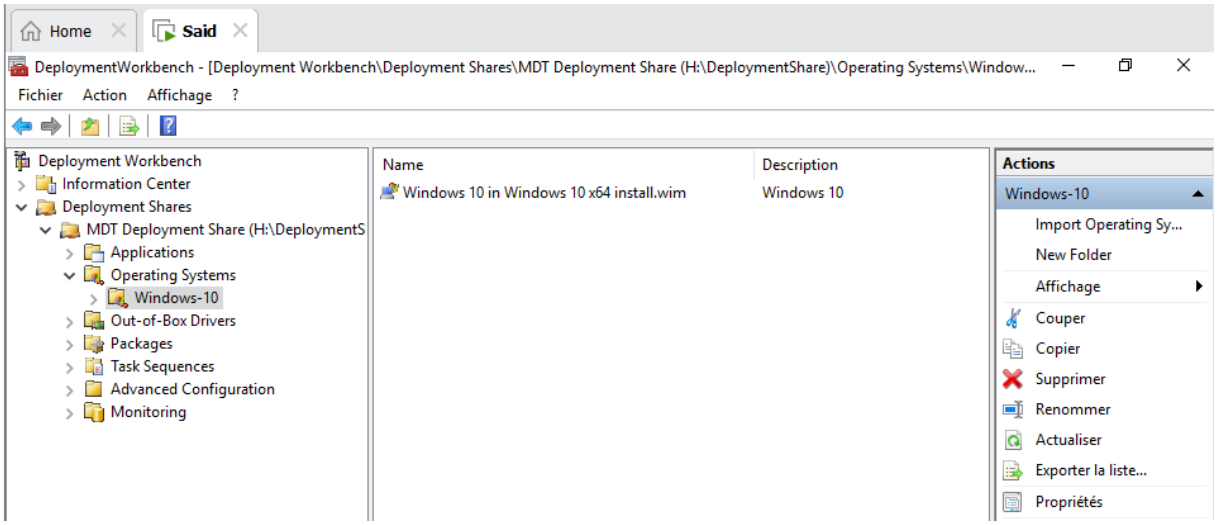


Déploiement

J'ai monté une image ISO de Windows 10 sur le lecteur CD/DVD. Je l'importe dans Deployment Workbench.



On a bien réussi à ajouter l'ISO de Windows 10 dans Deploment Share.

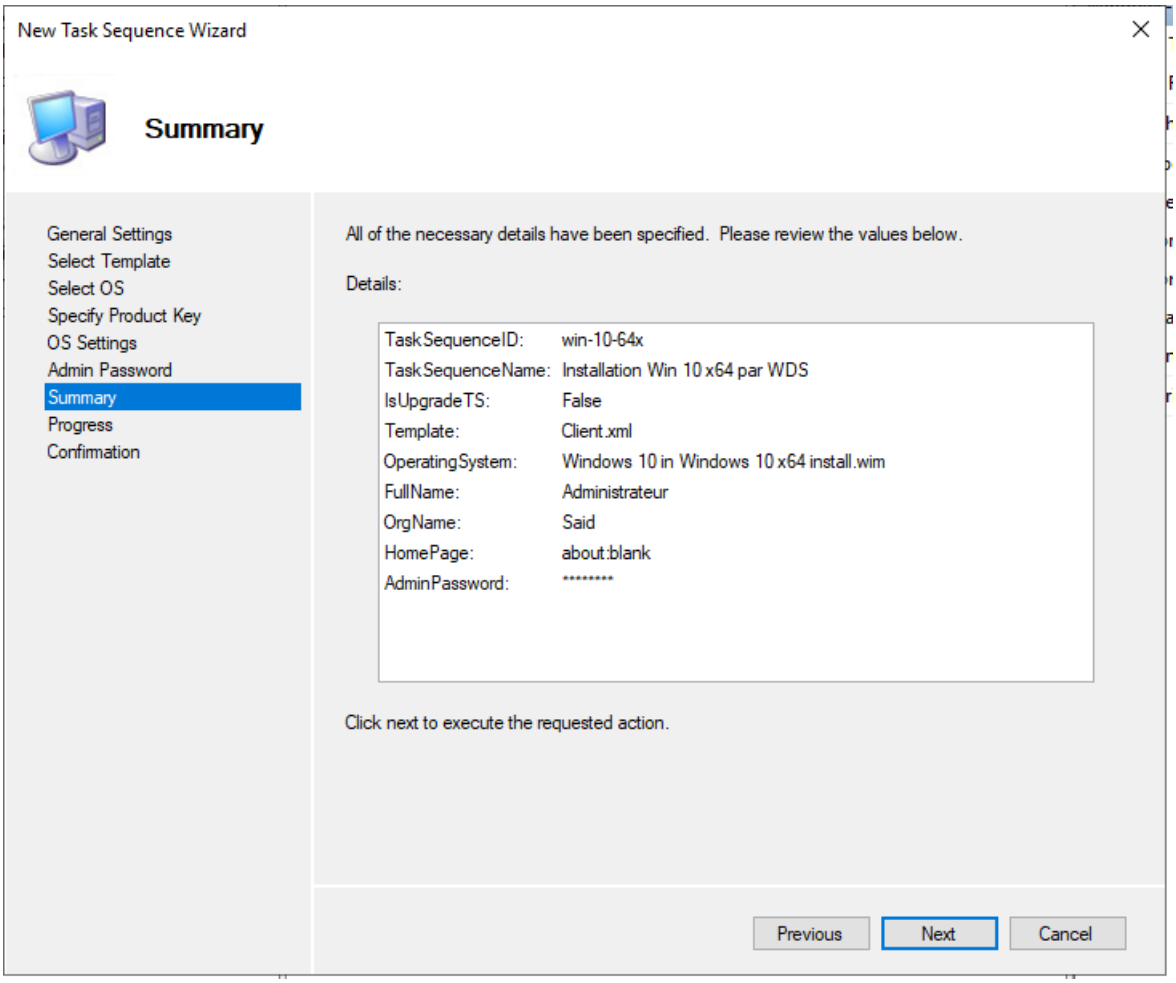


Création d’une tâche de séquence

Une tâche de séquence dans Deployment Share est une série d'étapes automatisées pour le déploiement d'une image système sur une machine cliente. Pour la créer :

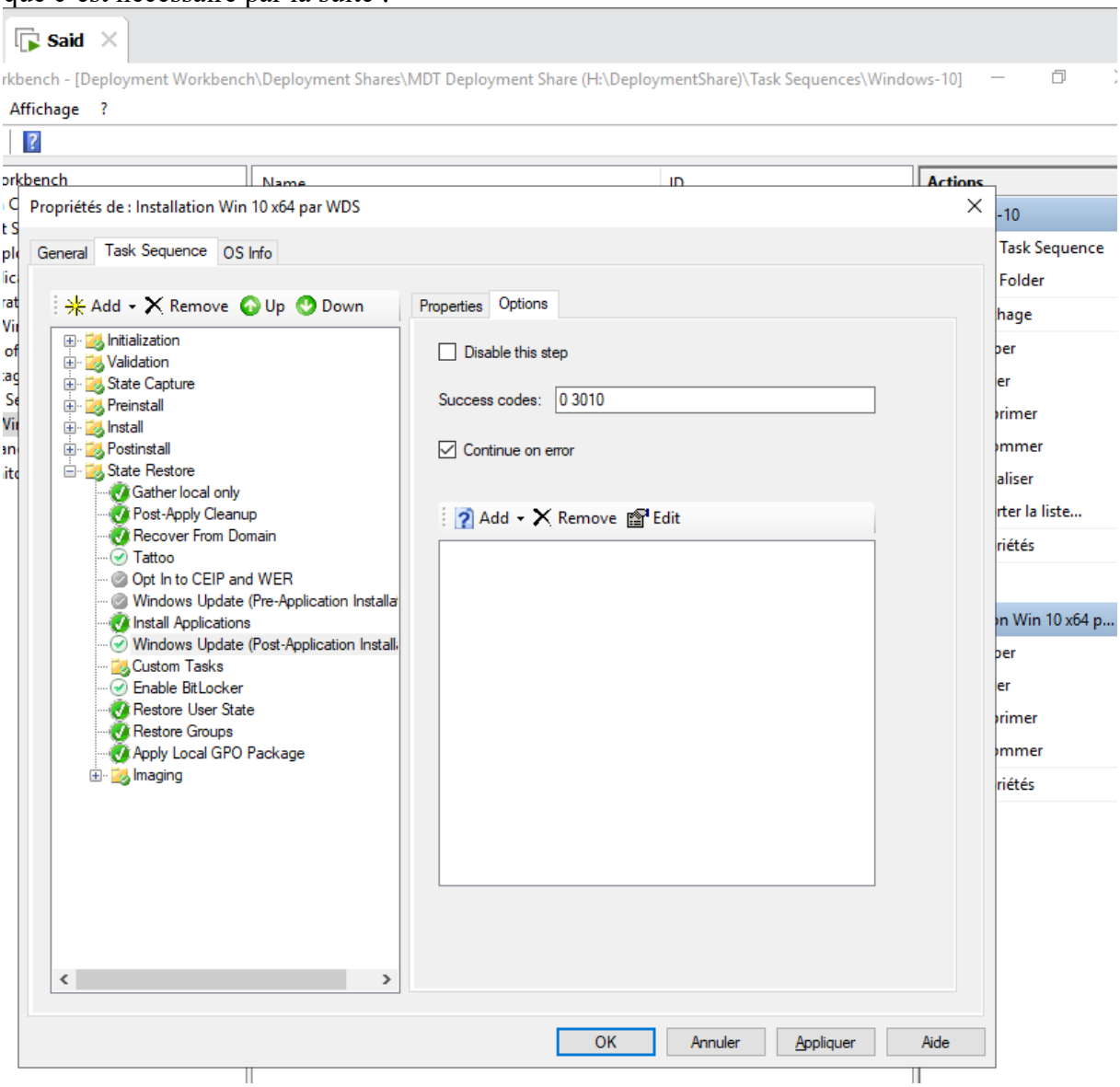
1. Dans la console Deployment Workbench, sous "Deployment Share", cliquez avec le bouton droit sur "Task Sequences" et sélectionnez "New Task Sequence".
2. Suivez l'assistant pour nommer la tâche, choisir l'image à déployer et configurer les paramètres.
3. Une fois terminé, la tâche de séquence est prête à être déployée sur les machines clientes pour automatiser le processus de déploiement de l'image système.

Voici le récap des détails sélectionnées :



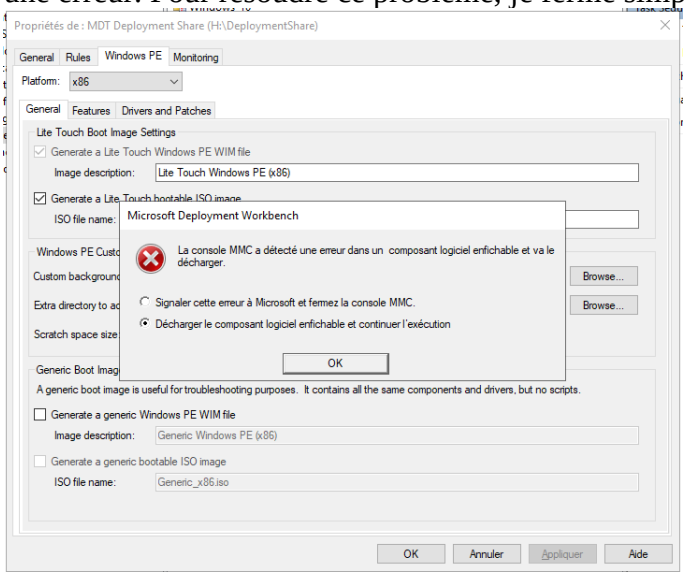
Je peux consulter les étapes de déploiement en allant sur :
Deployment Workbench > Deployment Share > Task Sequence > Win-10 > Propriétés

Dans les propriétés de la tâche de sequence je vais activer l'option « Windows Update » parce que c'est nécessaire par la suite :



Solution pour quelques erreurs existantes dans Microsoft MDT

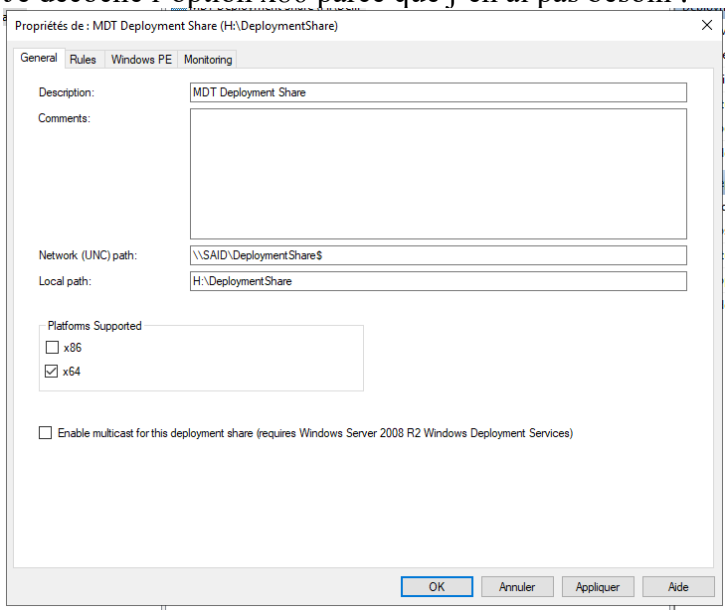
Lorsque j'accède aux propriétés de déploiement, puis à la fenêtre "Windows PE", je rencontre une erreur. Pour résoudre ce problème, je ferme simplement la console.



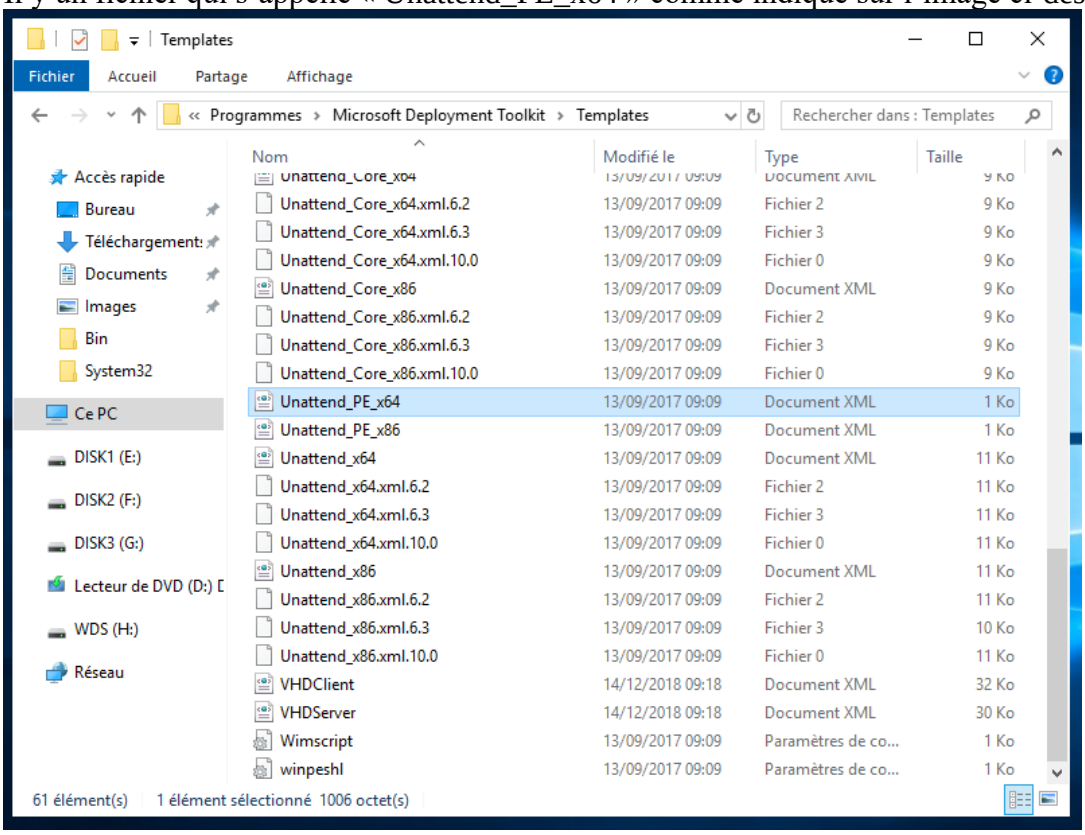
Pour résoudre ce problème, je vais écrire quelques lignes de commandes PowerShell :

```
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs
PS C:\Users\Administrateur> mkdir "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environme
nt\x86\WinPE_OCs"


    Répertoire: C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation
    Environment\x86


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        03/05/2024     13:58                WinPE_OCs


PS C:\Users\Administrateur>
```

Je décoche l'option x86 parce que j'en ai pas besoin :



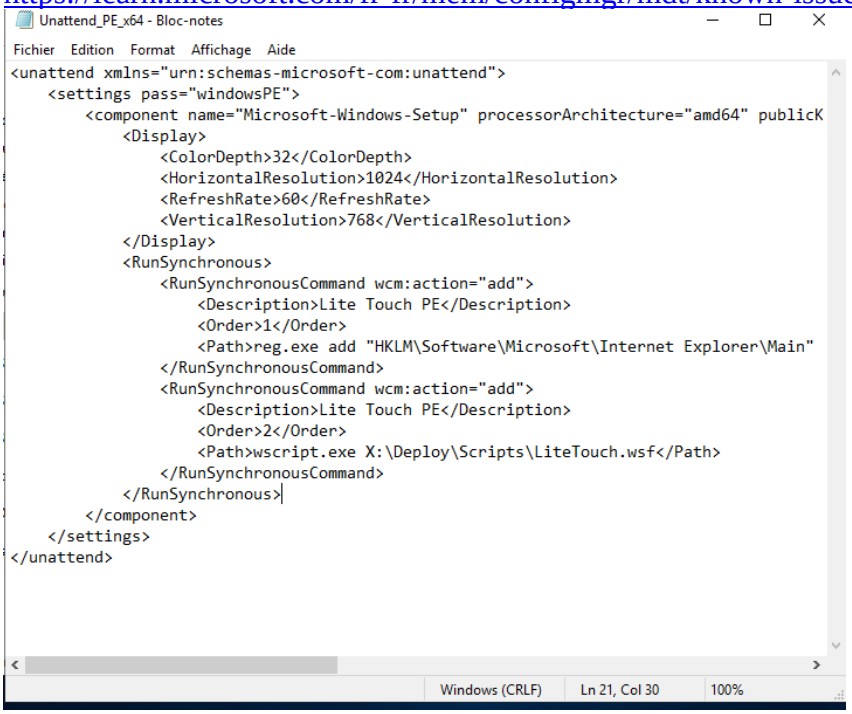
Il y un fichier qui s'appelle « Unattend_PE_x64 » comme indiqué sur l'image ci-dessous :



Je dois modifier ce fichier aussi, parce que ça peut nous causer des soucis, en premier temps, je crée une copie de ce fichier au cas où et ensuite je procède à la modification :

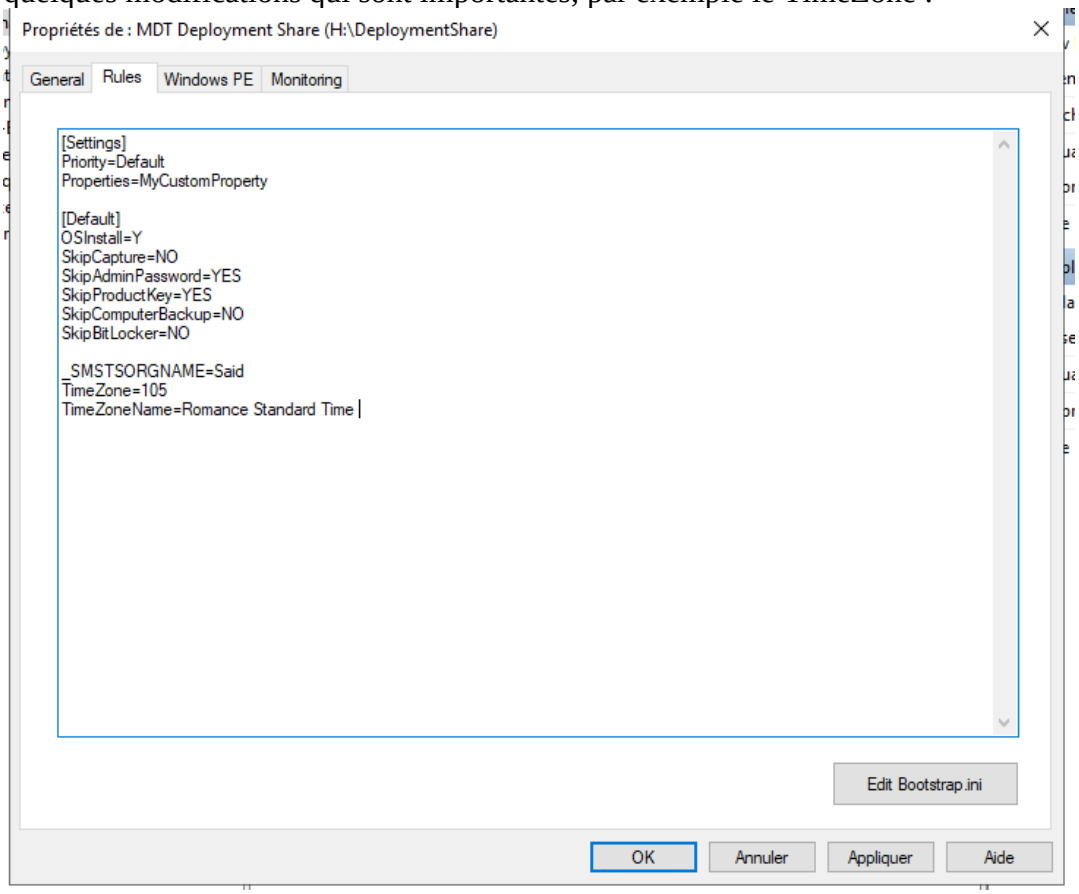
1- Je supprime tous qu'il y a dans ce fichier.
2- Je vais sur le lien ci-dessous pour avoir le contenu mis à jour pour ce fichier, je copie le contenu du site Microsoft, et je fais coller sur le fichier.

https://learn.microsoft.com/fr-fr/mem/configmgr/mdt/known-issues

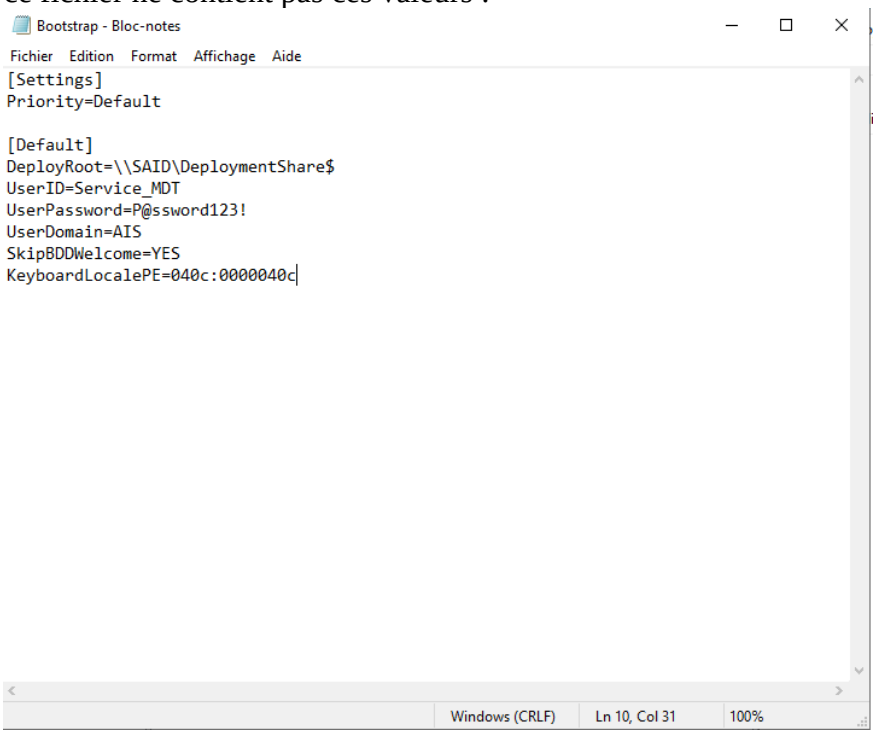


Quelques Modifications Utiles

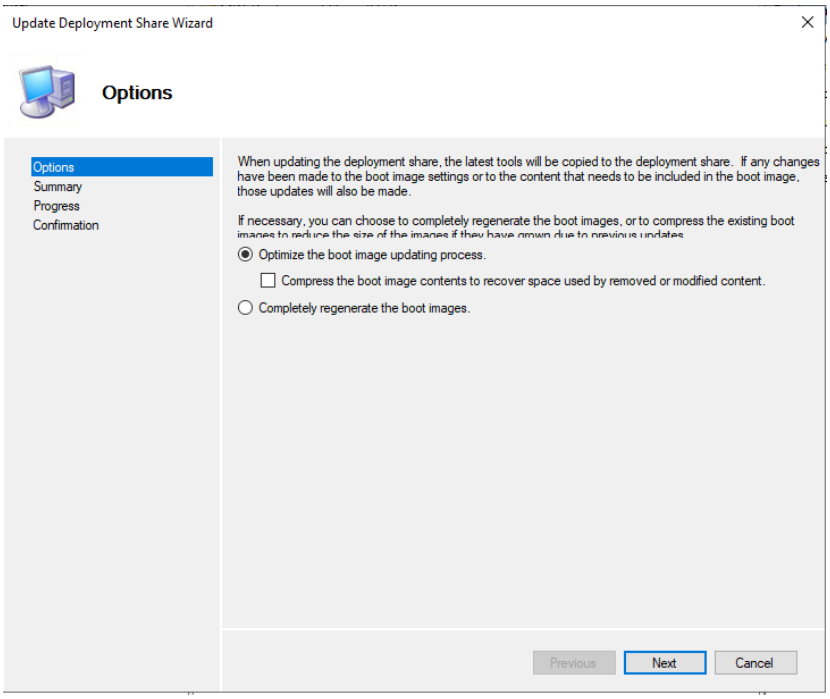
Je vais dans la rubrique propriétés de MDT Deployment Share, ensuite sur Rules j'effectue quelques modifications qui sont importantes, par exemple le TimeZone :



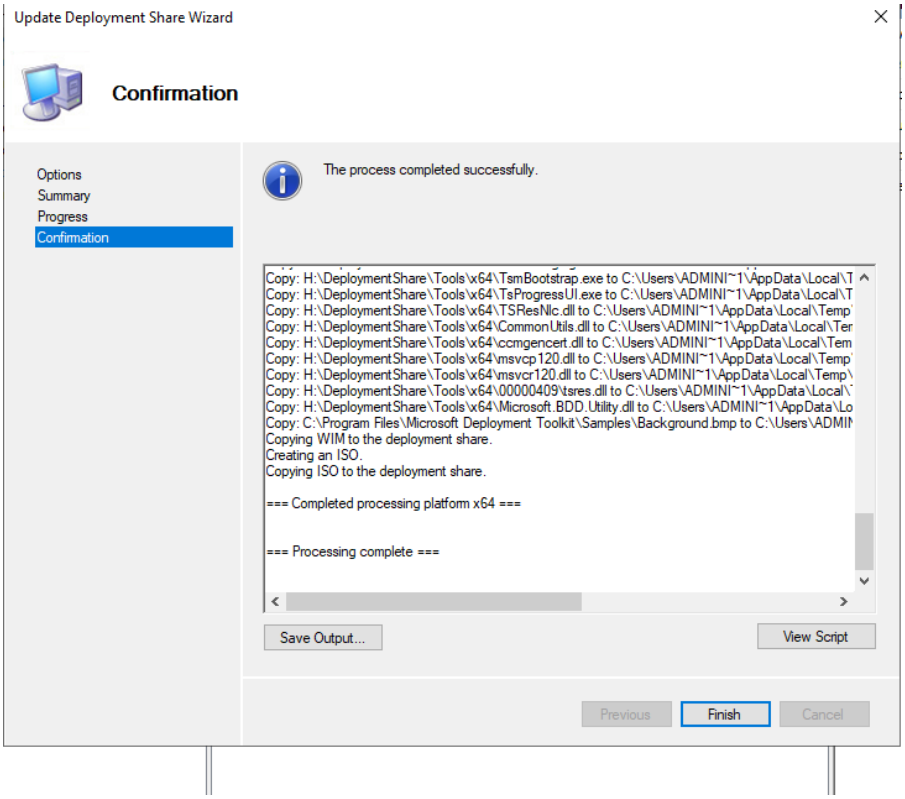
Il y une 2e modification à faire sur le fichier Bootstrap.ini qui affiché sur l'image en haut.
Il faut bien enregistrer le login et le mot de passe du service_MDT sur ce fichiers, par défaut, ce fichier ne contient pas ces valeurs :



Il faut bien vérifier que sur la partie de propriétés > Windows PE > l'option x64 est coché et x86 est décoché (Parce qu'on n'en pas besoin).
Ensuite il faut faire clique droit sur MDT Deployment Share et cliquer sur Update Delpoyment Share ; on sélectionne « Optimize the boot Image Updating Process » et on clique sur « Next » :

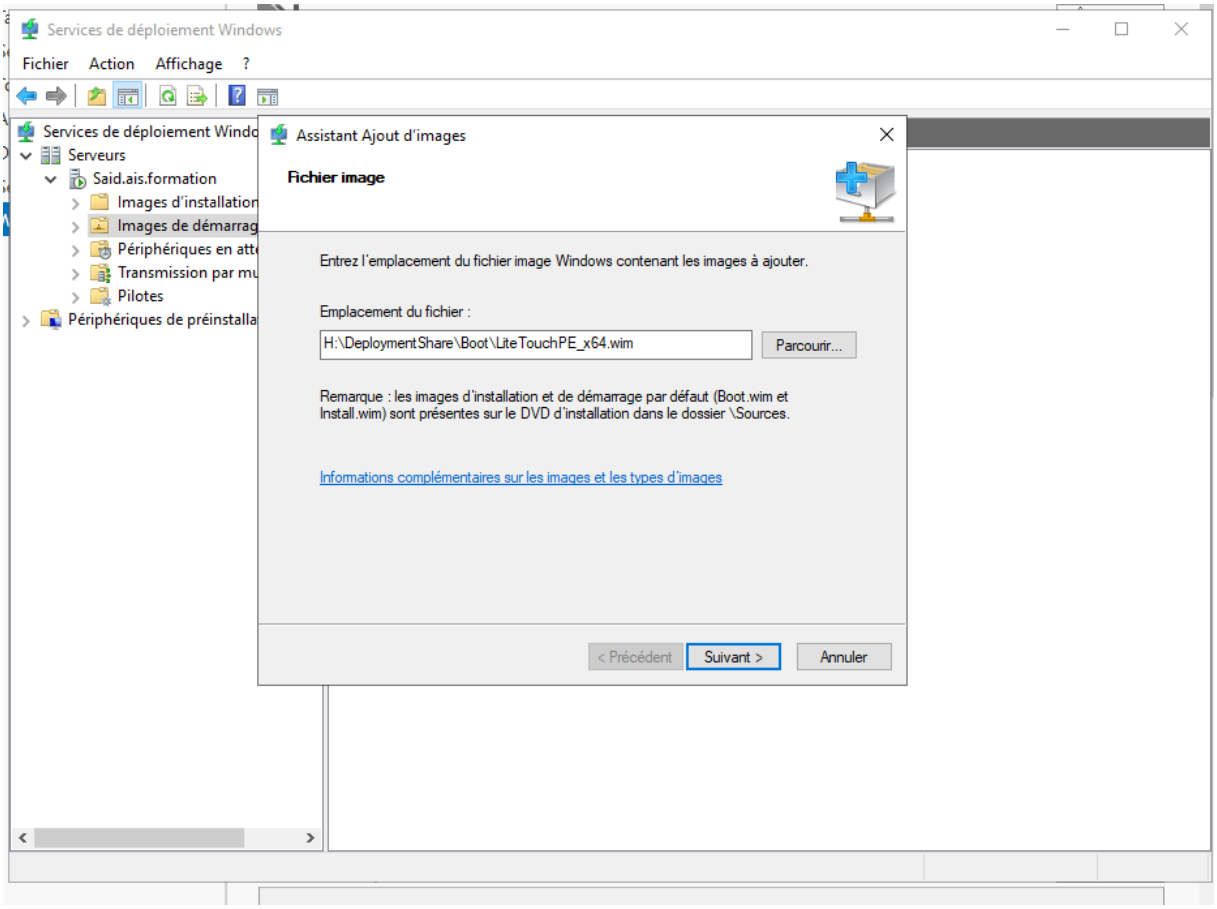


Ensuite ça crée l'image .Wim de Windows, on vérifie bien qu'il n'y a pas d'erreur dans l'exécution :

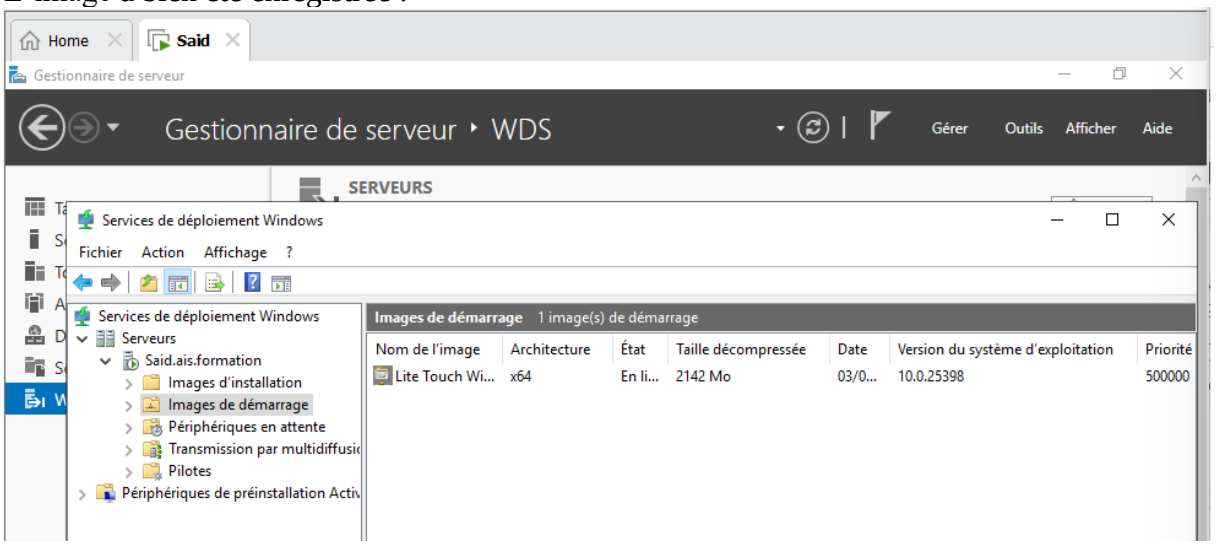


Manipulation sur le service WDS

Après la configuration du service WDS, je clique sur « Images de démarrage » et je détermine le chemin d'accès au fichier. Wim :

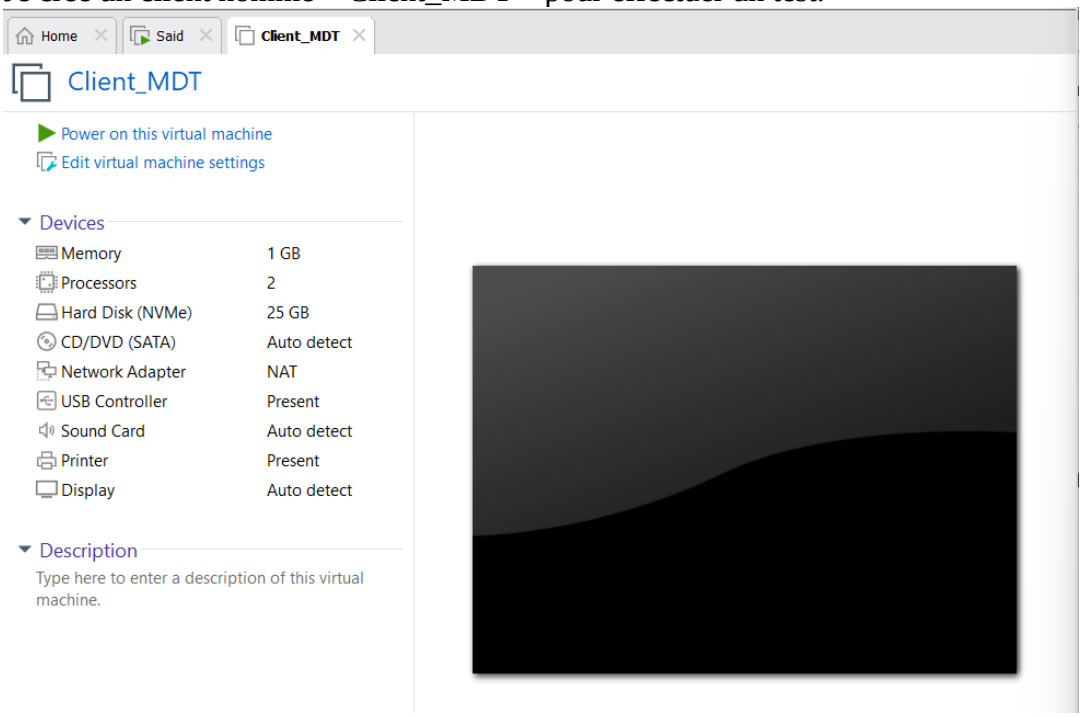


L'image a bien été enregistrée :

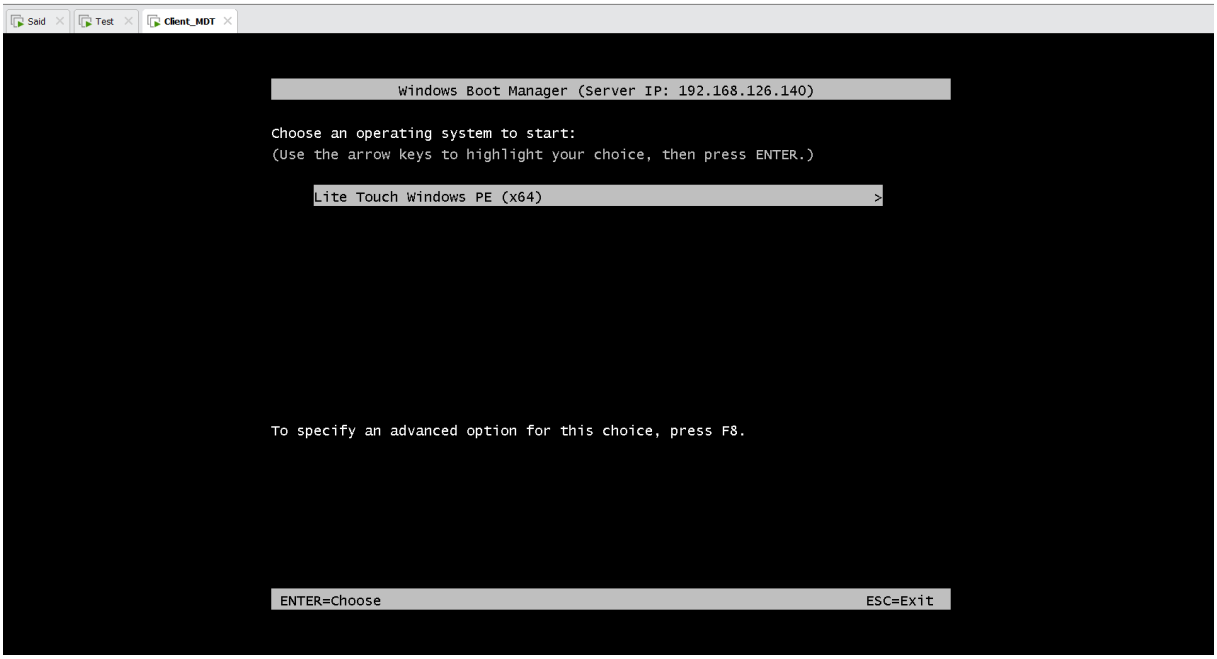


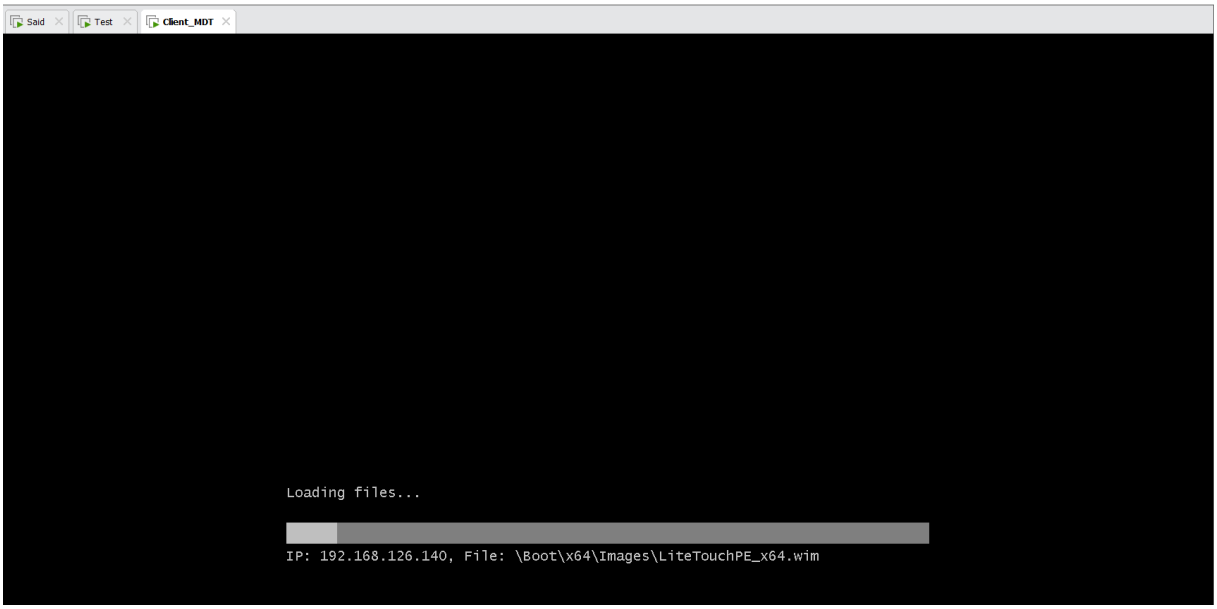
Installation de Windows 10 avec FileZilla sur le PC Client

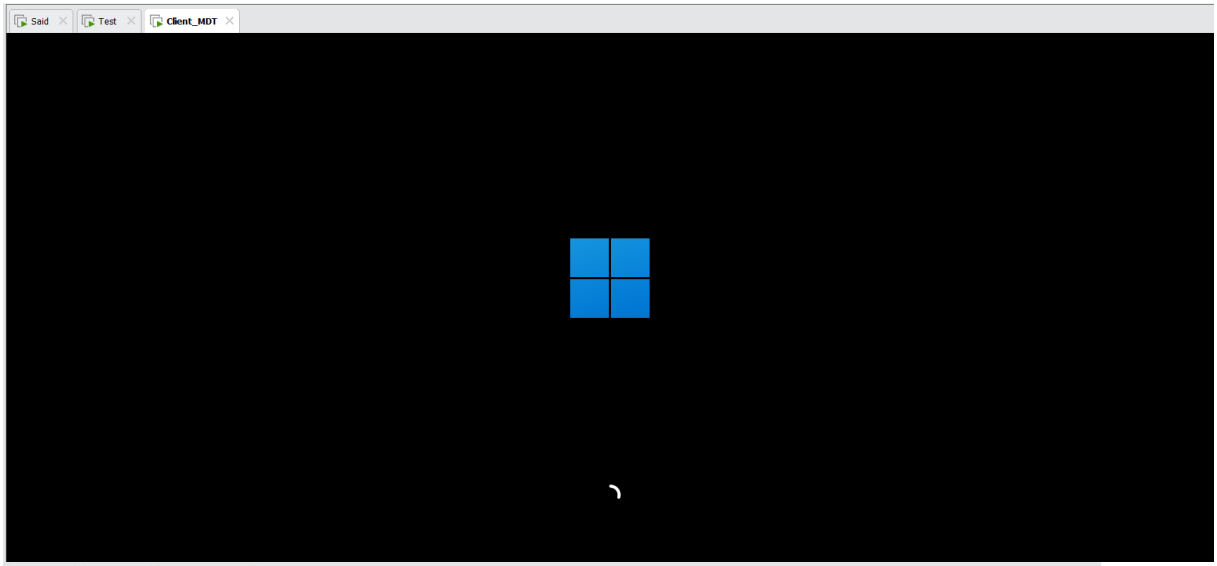
Je crée un client nommé « Client_MDT » pour effectuer un test.

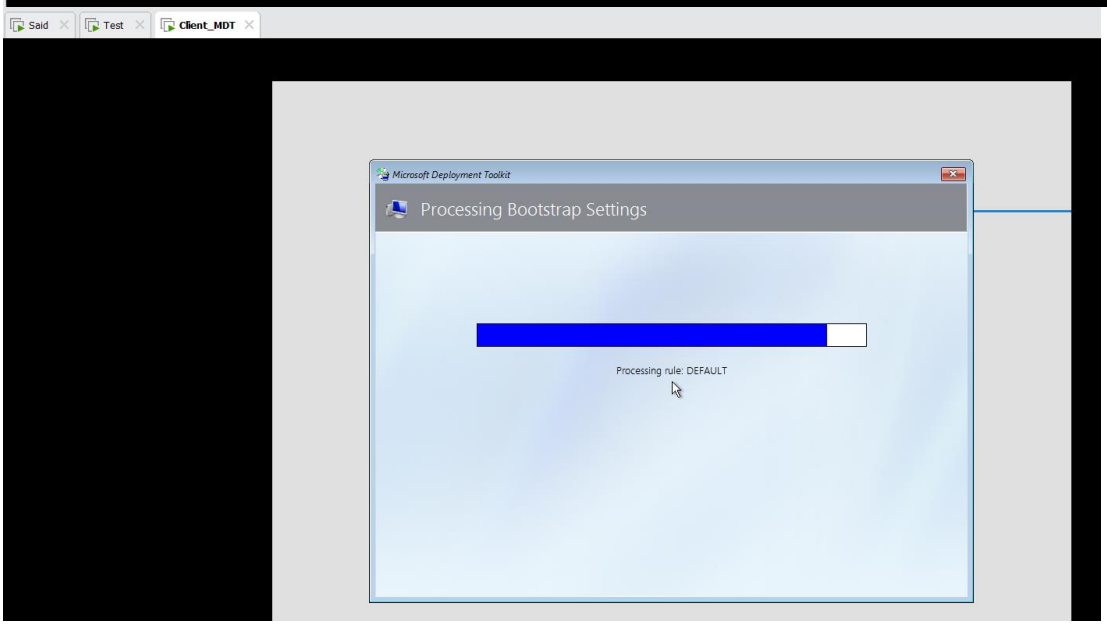


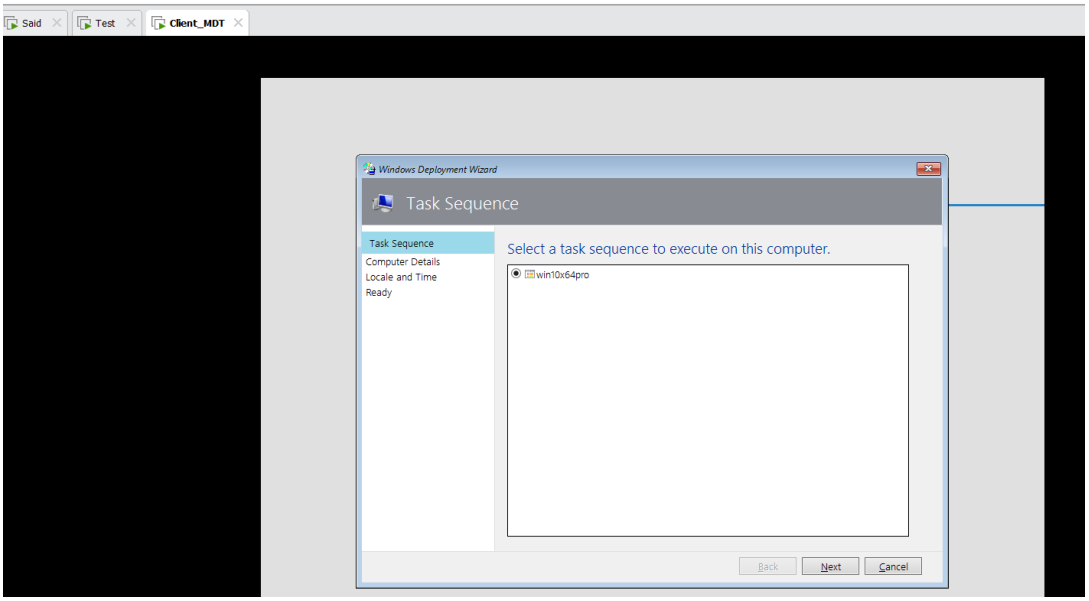
J'ai démarré la machine virtuelle et appuyé sur F12 pour Network Boot. L'installation a commencé tout de suite.

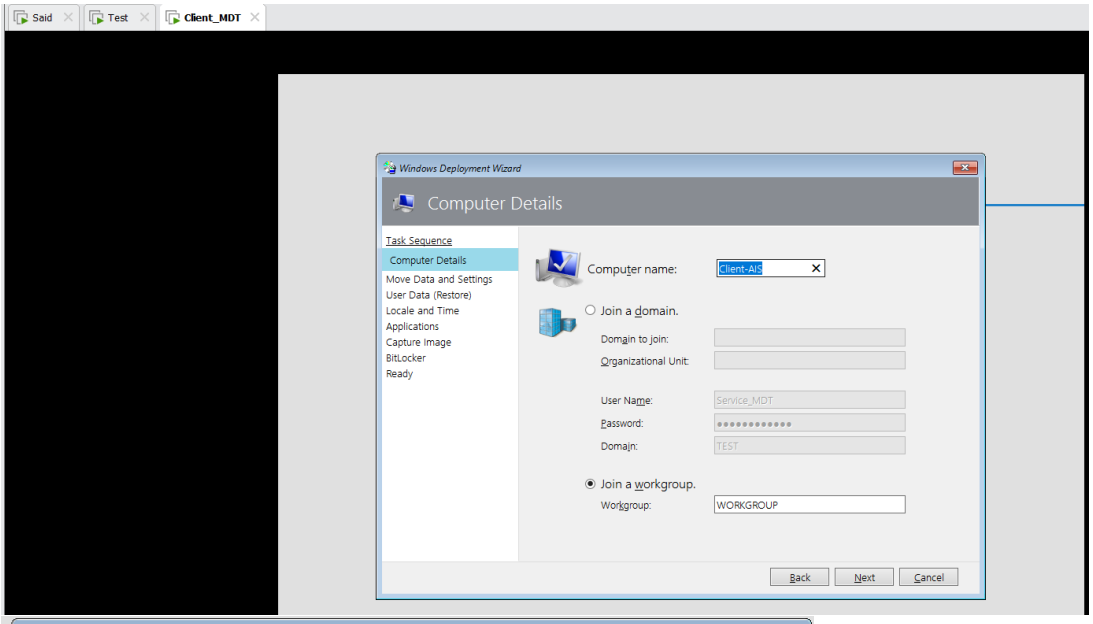


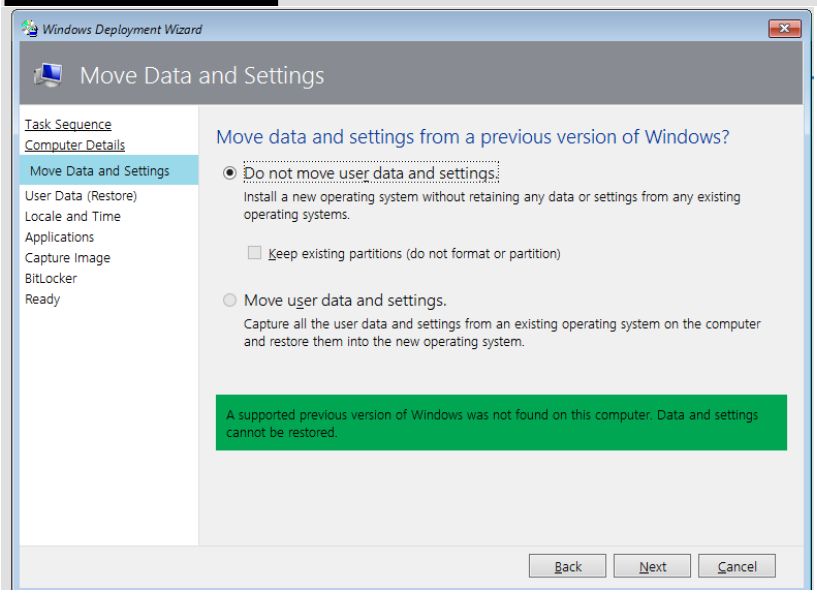

Said
Test
Client_MDT
Loading files...
IP: 192.168.126.140, File: \Boot\x64\Images\LiteTouchPE_x64.wim

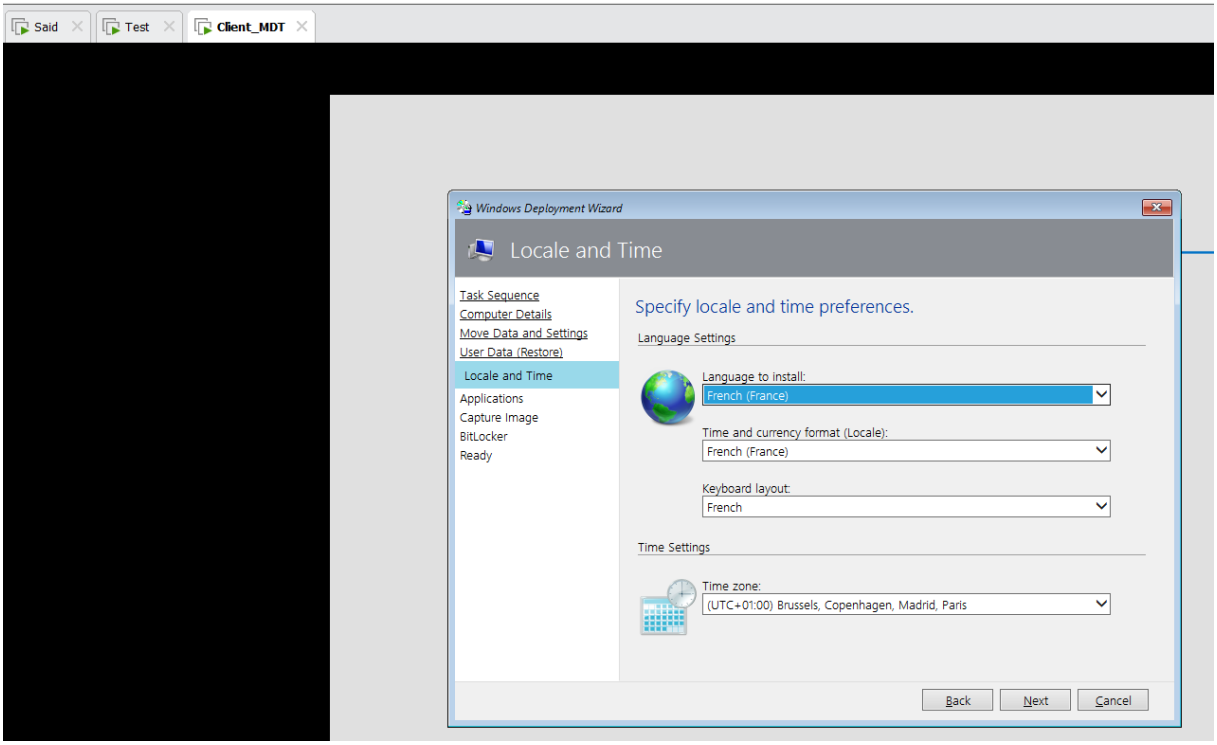

Said
Test
Client_MDT

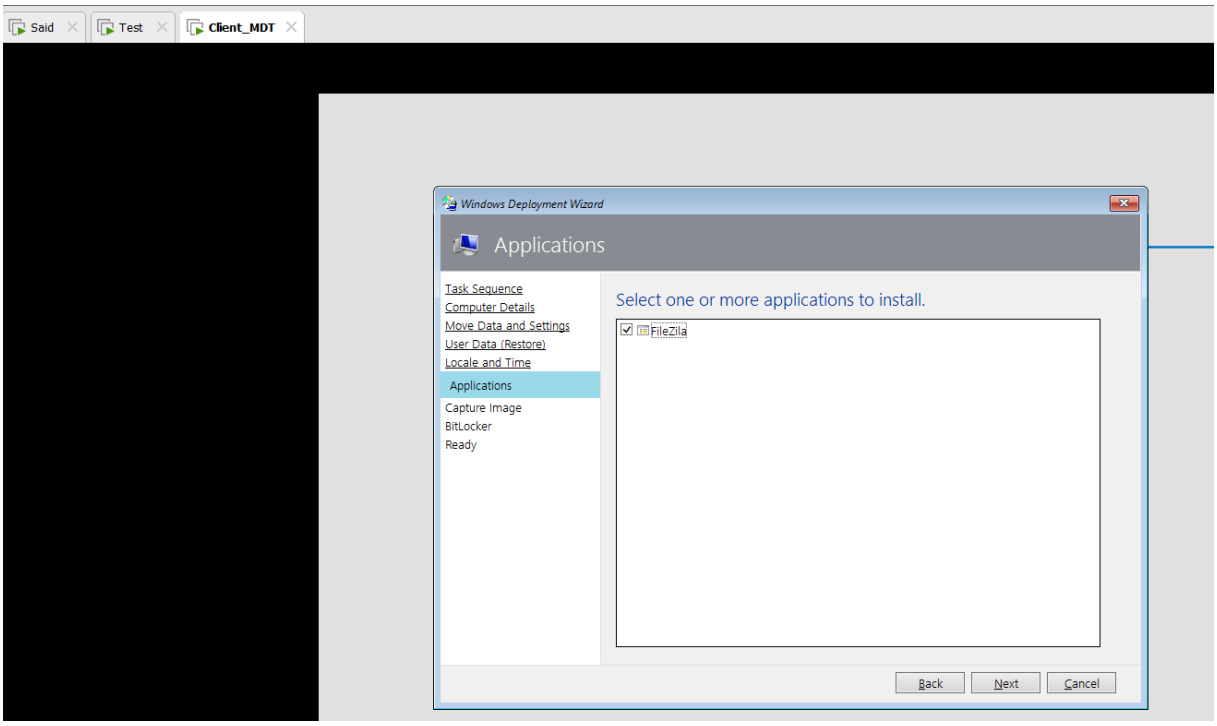

Said
Test
Client_MDT
Microsoft Deployment Toolkit
Processing Bootstrap Settings
Processing rule: DEFAULT

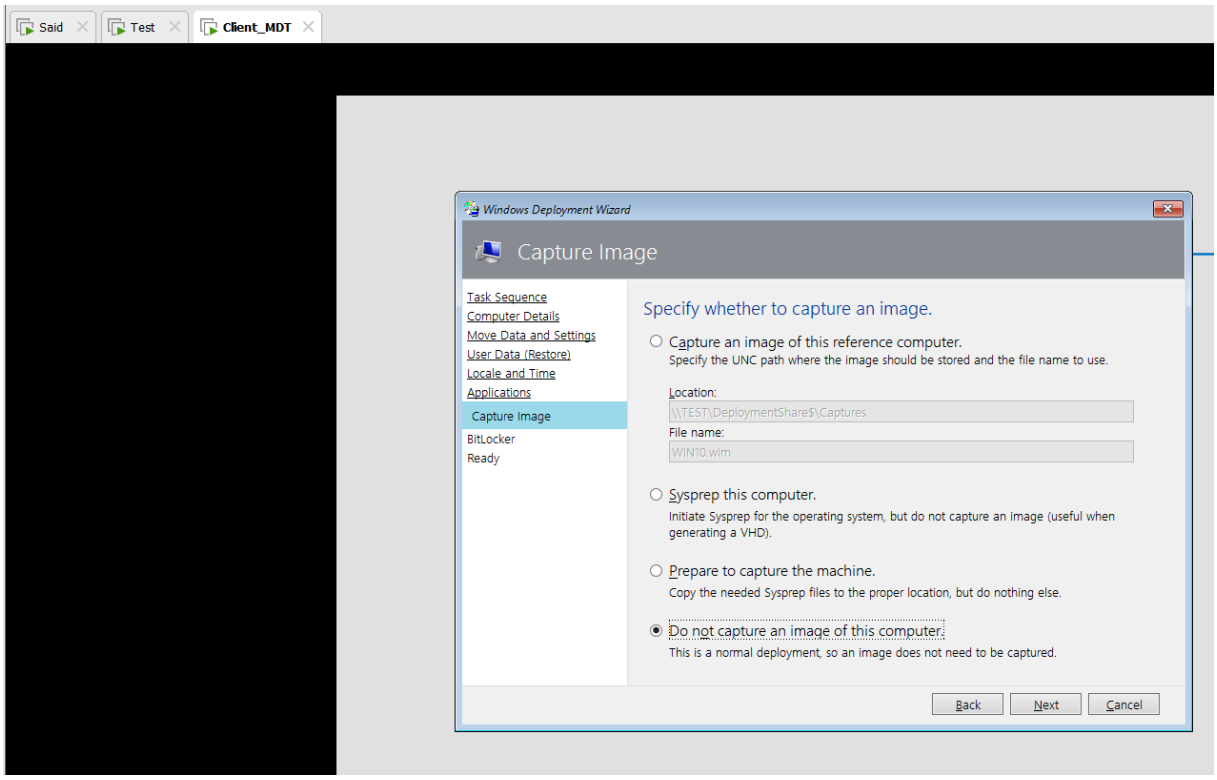

Said
Test
Client_MDT
Windows Deployment Wizard
Task Sequence
Task Sequence
Computer Details
Locale and Time
Ready
Select a task sequence to execute on this computer.
win10x64pro
Back
Next
Cancel

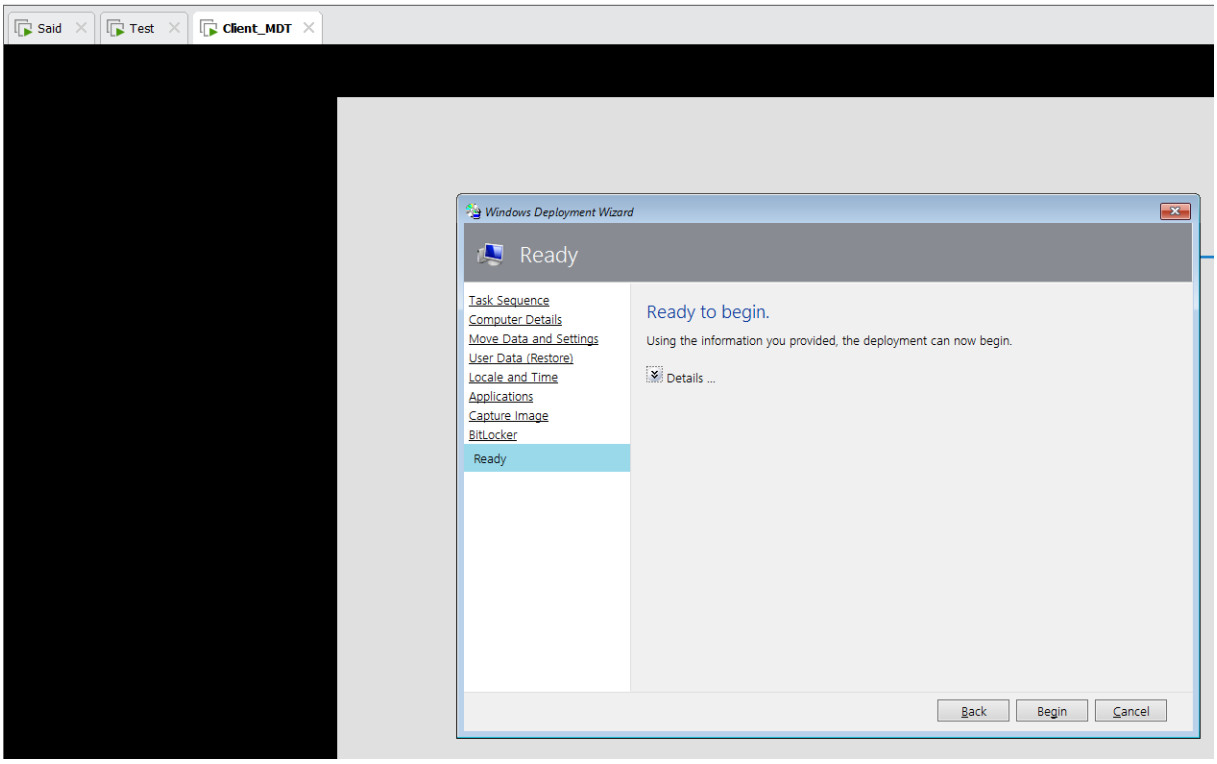

Said
Test
Client_MDT
Windows Deployment Wizard
Computer Details
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Computer name:
Client-AIS
Join a domain.
Domain to join:
Organizational Unit:
User Name:
Service_MDT
Password:
Domain:
TEST
Join a workgroup.
Workgroup:
WORKGROUP
Back
Next
Cancel

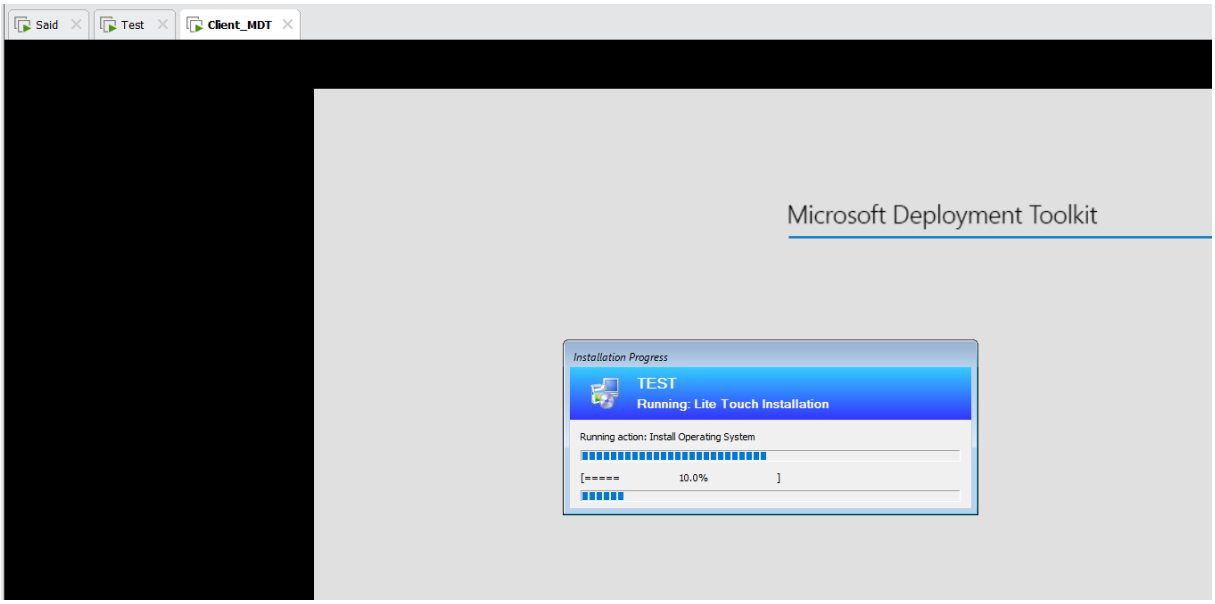

Windows Deployment Wizard
Move Data and Settings
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Move data and settings from a previous version of Windows?
Do not move user data and settings.
Install a new operating system without retaining any data or settings from any existing operating systems.
Keep existing partitions (do not format or partition)
Move user data and settings.
Capture all the user data and settings from an existing operating system on the computer and restore them into the new operating system.
A supported previous version of Windows was not found on this computer. Data and settings cannot be restored.
Back
Next
Cancel


Said
Test
Client_MDT
Windows Deployment Wizard
Locale and Time
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Specify locale and time preferences.
Language Settings
Language to install:
French (France)
Time and currency format (Locale):
French (France)
Keyboard layout:
French
Time Settings
Time zone:
(UTC+01:00) Brussels, Copenhagen, Madrid, Paris
Back
Next
Cancel


Said
Test
Client_MDT
Windows Deployment Wizard
Applications
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Select one or more applications to install.
FileZilla
Back
Next
Cancel


Said
Test
Client_MDT
Windows Deployment Wizard
Capture Image
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Specify whether to capture an image.
Capture an image of this reference computer.
Specify the UNC path where the image should be stored and the file name to use.
Location:
\\TEST\DeploymentShare$\Captures
File name:
WIN10.wim
Sysprep this computer.
Initiate Sysprep for the operating system, but do not capture an image (useful when generating a VHD).
Prepare to capture the machine.
Copy the needed Sysprep files to the proper location, but do nothing else.
Do not capture an image of this computer.
This is a normal deployment, so an image does not need to be captured.
Back
Next
Cancel

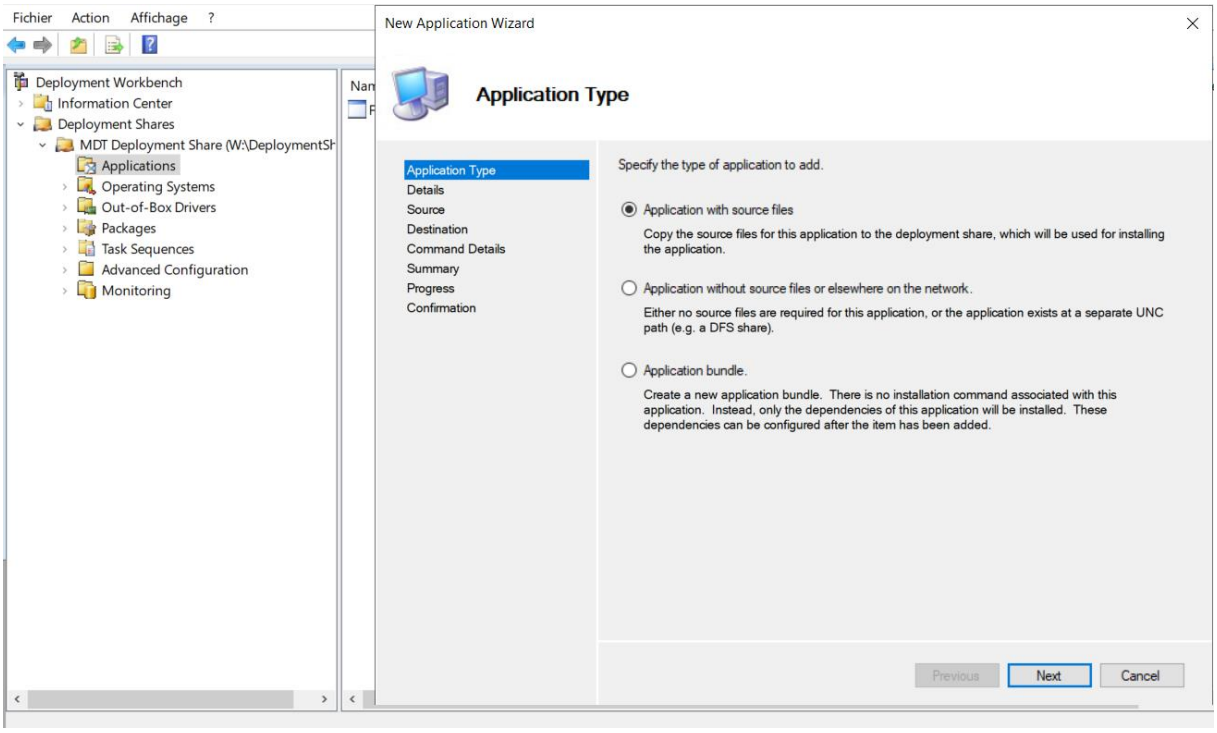

Said
Test
Client_MDT
Windows Deployment Wizard
Ready
Task Sequence
Computer Details
Move Data and Settings
User Data (Restore)
Locale and Time
Applications
Capture Image
BitLocker
Ready
Ready to begin.
Using the information you provided, the deployment can now begin.
Details ...
Back
Begin
Cancel


Said
Test
Client_MDT
Microsoft Deployment Toolkit
Installation Progress
TEST
Running: Lite Touch Installation
Running action: Install Operating System
[===== 10.0%]

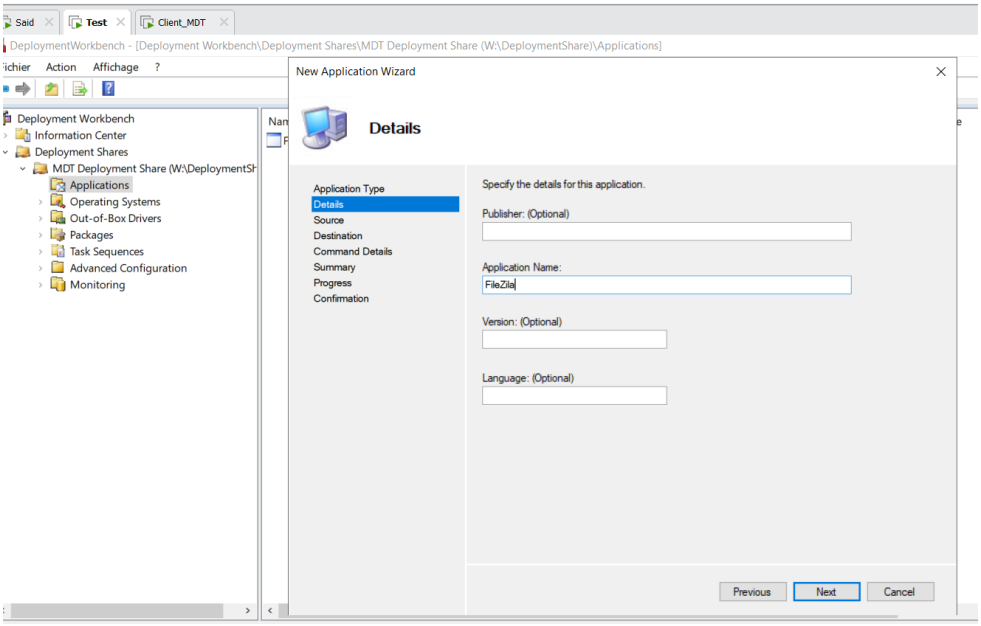
Déploiement de FileZilla via MDT

Par cette méthode (MDT) je peux installer des applications, en même temps que l'OS. On procède à cette étape toute suite.

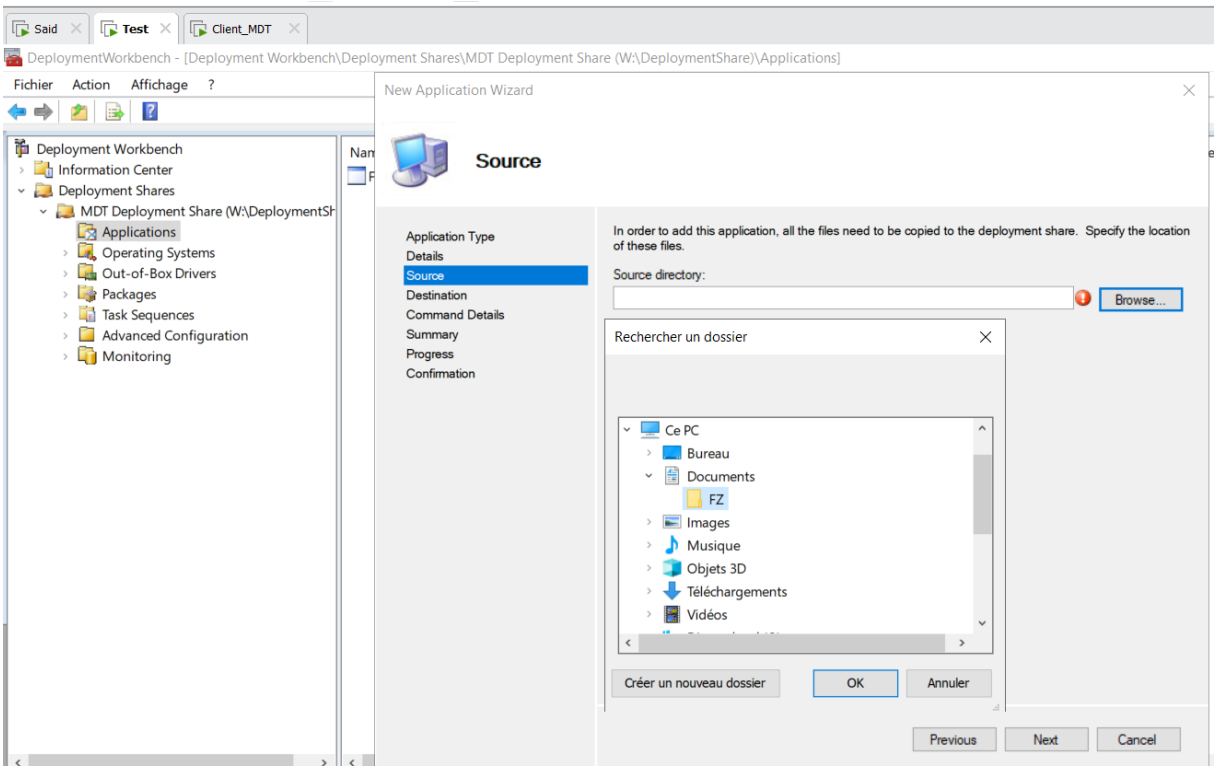
Dane le Deployment Workbench, on clique sur "Applications" dans la fenêtre qui s'affiche on va cliquer sur « Application with source files » puis suivant.



Ensuite on écrit le nom d'application qu'on veut installer, dans ce cas, FileZila :

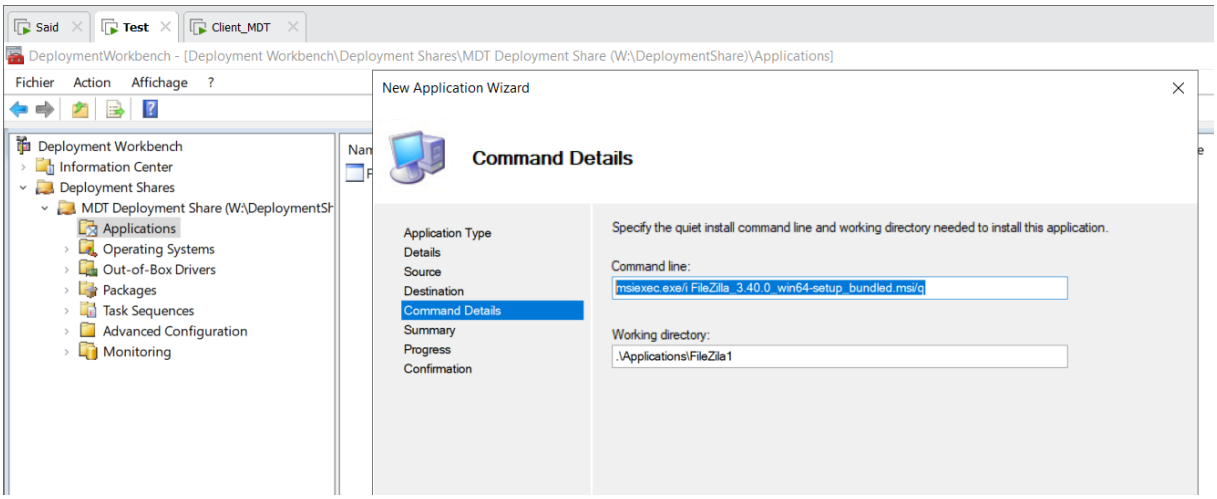


A cette étape, on va sélectionner l'emplacement du fichier MSI :

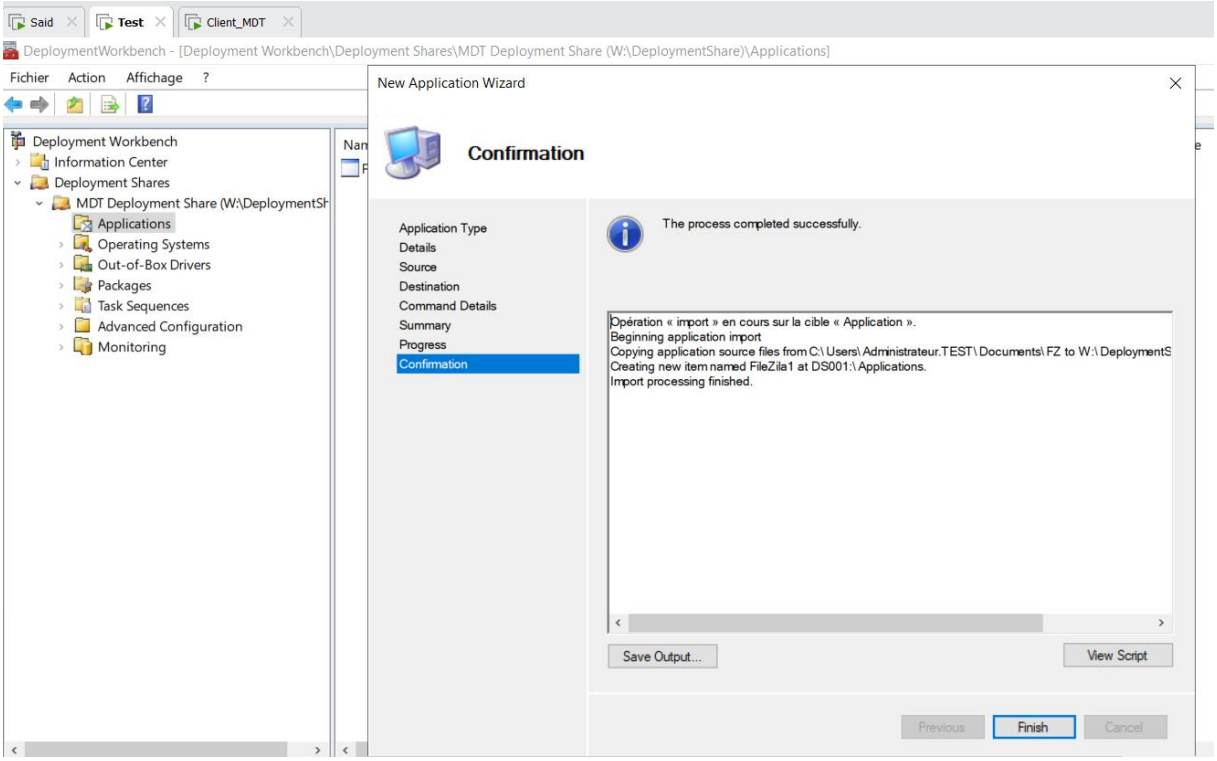


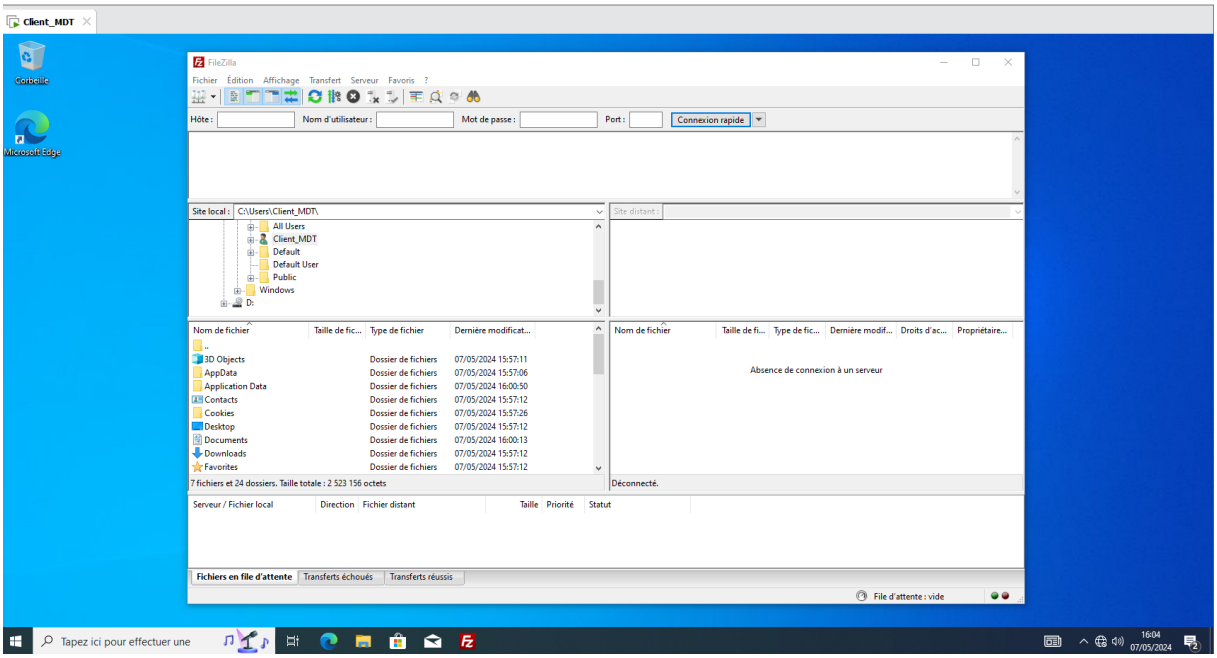
Après cette étape, on doit écrire la commande suivante puis on clique sur suivant.

msiexec.exe/i FileZilla_3.40.0_win64-setup_bundled.msi/q



On a bien réussi à charger l'application sur deployment workbench pour être installée.



Résultat

Dans ce projet, j'ai réussi à installer Windows 10, ainsi que l'application FileZilla, en utilisant la méthode MDT ADK. Cette approche nous permet d'installer les applications en même temps que le système d'exploitation. L'efficacité de cette méthode est remarquable, mais elle requiert des compétences techniques spécifiques. Vous trouverez ci-dessous l'image du système d'exploitation et des applications installées.